Министерство образования Республики Беларусь

Учреждение образования

«Брестский Государственный технический университет»

Кафедра ИИТ

Лабораторная работа №2

По дисциплине «МРЗИС»

Тема: “Выявление аномалий”

Выполнил:

Студент 3 курса

Группы ИИ-21

Романко Н.А.

Проверил:

Туз И.С.

Брест 2023

Цель:реализовать программу способную выявлять аномалии на заданных данных

Метод: кластерный метод

Данные: Генерация датасета в виде нормального распределенных данных с некоторым количеством случайно добавленных значений

Код программы:

import numpy as np

import pandas as pd

import csv

import matplotlib.pyplot as plt

dataX = np.random.normal(5, 0.8, 5000)

dataX = dataX[dataX > 0]

dataY = np.random.normal(5, 0.8, 5000)

dataY = dataY[dataY > 0]

with open('MRZIS\\NotRandomData.csv', 'w', newline='') as csvfile:

    csvwriter = csv.writer(csvfile)

    csvwriter.writerow(['ValueX', 'ValueY'])

    for x, y in zip(dataX, dataY):

        csvwriter.writerow([x, y])

random\_data = np.random.uniform(0.1, 10, 200)

data\_with\_random = np.concatenate((dataX, random\_data), axis=None)

np.random.shuffle(data\_with\_random)

with open('MRZIS\\randomData.csv', 'w', newline='') as csvfile:

    csvwriter = csv.writer(csvfile)

    csvwriter.writerow(['ValueX', 'ValueY'])

    for value in data\_with\_random:

        # Генерируем случайное значение для y

        y = np.random.normal(5, 0.8)

        csvwriter.writerow([value, y])

not\_random\_data = pd.read\_csv('MRZIS\\NotRandomData.csv')

data = pd.read\_csv('MRZIS\\randomData.csv')

features = ["ValueX", "ValueY"]

data = data.dropna(subset=features)

#опредление кластеров

def random\_centroids(data, k):

    centroids = []

    for \_ in range(k):

        centroid = data.apply(lambda x: float(x.sample()))

        centroids.append(centroid)

    return pd.concat(centroids, axis=1)

#нахождение растояния между значением и кластерами

def get\_labels(data, centroids):

    distances = centroids.apply(lambda x: np.sqrt(((data - x)\*\*2).sum(axis=1)))

    return distances.idxmin(axis=1)  #какое число к какому кластеру идет

#пересчет центроидов

def new\_centroids(data, labels, k):

    return data.groupby(labels).apply(lambda x: np.exp(np.log(x).mean())).T

max\_iteration = 100

iteration = 1

centroid\_count = 3

anomaly\_threshold = 1.8

centroids = random\_centroids(data, centroid\_count)

old\_centroids = pd.DataFrame()

#изменение центроидов

while iteration < max\_iteration and not centroids.equals(old\_centroids):

    old\_centroids = centroids

    labels = get\_labels(data, centroids)

    centroids = new\_centroids(data, labels, centroid\_count)

    #plot\_clusters(data, labels, centroids, iteration)

    iteration += 1

data['Label'] = labels

data['DistanceToCentroid'] = np.sqrt(((data[['ValueX', 'ValueY']] - centroids.loc[:, labels].values.T) \*\* 2).sum(axis=1))

anomalies = data[data['DistanceToCentroid'] > anomaly\_threshold]

colors = ['b', 'g', 'y']

plt.figure(figsize=(8, 4))

plt.title("С кластеризацией и обнаружением аномалий")

plt.scatter(data['ValueX'], data['ValueY'], color=[colors[i] for i in labels], s=5)

plt.scatter(anomalies['ValueX'], anomalies['ValueY'], color='red', s=15, label='Аномалии')

plt.xlabel("ValueX")

plt.ylabel("ValueY")

plt.legend()

plt.show()

Начальные данные:

Изображение выглядит как снимок экрана, текст, программное обеспечение

Автоматически созданное описание

Данные с выявлением аномалий:

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Вывод: реализовал программу для выявления аномалий на заданном датасете при помощи кластерного метода