Министерство образования Республики Беларусь

Учреждение образования

«Брестский Государственный технический университет»

Кафедра ИИТ

**Лабораторная работа №2**

По дисциплине «Модели решения задач в интеллектуальных системах»

Тема: «Выявление аномалий»

**Выполнил:**

Студент 3 курса

Группы ИИ-21

Литвинюк Т. В.

**Проверил:**

Туз И. С.

Брест 2023

**Цель:** реализовать программу, способную выявлять аномалии на заданных данных.

**Ход работы**

**Вариант 1**

**Метод:** Машина опорных векторов для одного класса

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

*from* matplotlib *import* pyplot *as* plt

*import* numpy *as* np

*# Генерация данных (нормальное распределение)*

np.random.seed(0)

mean = [0, 0]  *# Средние значения*

cov = [[1, 0.5], [0.5, 1]]  *# Ковариационная матрица*

num\_samples = 200  *# Количество нормальных данных*

normal\_data = np.random.multivariate\_normal(mean, cov, num\_samples)

*# Параметр nu контролирует долю аномалий, его можно настроить*

nu = 0.05

*# Обучение One-Class SVM*

def train\_one\_class\_svm(normal\_data, nu):

    x\_, y\_ = [x[0] *for* x *in* normal\_data], [y[1] *for* y *in* normal\_data]

    plt.scatter(x\_, y\_)

    plt.show()

    num\_samples, num\_features = normal\_data.shape

*# Вычисление среднего и дисперсии для каждой функции*

    mean = np.mean(normal\_data, axis=0)

    variance = np.var(normal\_data, axis=0)

*# Ширина окна (сглаживающий параметр)*

    width = 2.0 \* np.sqrt(variance)

*# Преобразование данных*

    normal\_data\_scaled = (normal\_data - mean) / width

*# Построение матрицы Грама*

    gram\_matrix = np.dot(normal\_data\_scaled, normal\_data\_scaled.T)

*# Вычисление bias*

    bias = -np.percentile(gram\_matrix, 100.0 \* nu)

*return* mean, width, bias

mean, width, bias = train\_one\_class\_svm(normal\_data, nu)

*# Оценка новых данных*

def predict\_anomalies(data, mean, width, bias):

    data\_scaled = (data - mean) / width

    gram\_matrix = np.dot(data\_scaled, data\_scaled.T)

    decision\_function = gram\_matrix + bias

*return* decision\_function

*# Генерация новых данных для оценки*

new\_data = np.random.multivariate\_normal(mean, cov, 10)  *# 10 новых данных*

*# Добавление аномальных данных*

*for* \_ *in* range(5):

    new\_data = np.vstack((new\_data, np.random.random(2) \* 10))

np.random.shuffle(new\_data)

anomalies = predict\_anomalies(new\_data, mean, width, bias)

anomaly\_scores = np.mean(anomalies, axis=1)

boolean\_anomaly = anomaly\_scores > 3

*# Вывод графика датасета*

x\_, y\_ = [x[0] *for* x *in* new\_data], [y[1] *for* y *in* new\_data]

x, y, x\_anom, y\_anom= [], [], [], []

*for* i *in* range(len(x\_)):

*if* boolean\_anomaly[i]:

        x\_anom.append(x\_[i])

        y\_anom.append(y\_[i])

*else*:

        x.append(x\_[i])

        y.append(y\_[i])

plt.scatter(x\_, y\_, s=20)

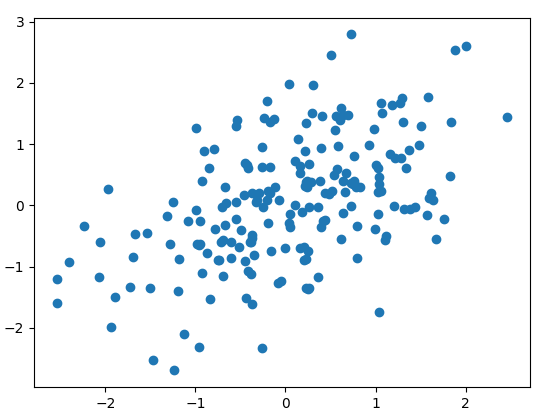
plt.scatter(x\_anom, y\_anom, s=20, c='r')

plt.show()

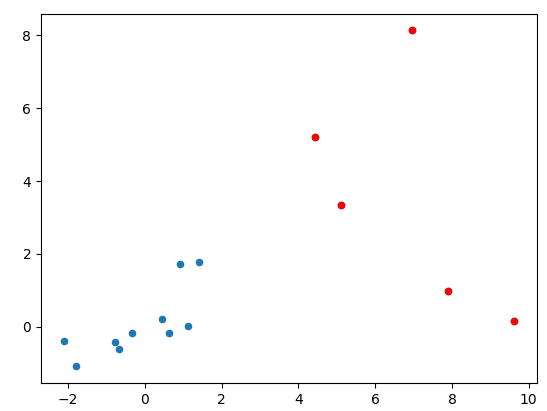
print("Аномальные оценки для новых данных:")

print(anomaly\_scores)

Данные, используемые для обучения:



Данные при тестировании:



Еще одним методом выявления аномалий является метод кластеризации методом DBSCAN (Density-Based Spatial Clustering of Applications with Noise). Он основан на плотности данных и позволяет выявлять аномалии как точки, которые не принадлежат ни одному кластеру или находятся в областях с низкой плотностью.

Ключевые особенности метода DBSCAN:

1. **Кластеризация на основе плотности:** DBSCAN определяет кластеры, исходя из плотности точек данных в окрестности каждой точки. Точки, которые находятся близко друг к другу и имеют достаточно близких соседей, объединяются в кластеры.
2. **Аномалии как шум:** Точки, которые не могут быть объединены ни с одним кластером и имеют недостаточно соседей, считаются аномалиями (шумом).
3. **Параметры:** DBSCAN имеет два основных параметра: eps (радиус окрестности) и min\_samples (минимальное количество точек в окрестности для формирования кластера). Их настройка может влиять на способность метода выявлять аномалии.

DBSCAN хорошо подходит для выявления аномалий в данных, где аномалии образуют отдельные кластеры или находятся в областях с низкой плотностью. Он не требует заранее заданного числа кластеров и может обнаруживать кластеры различной формы.

**Вывод:** в ходе лабораторной работы я научился выявлять аномалии с помощью некоторых методов.