ГУАП

КАФЕДРА № 43

ОТЧЁТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| д-р техн. наук, доцент |  |  |  | С. И. Колесникова |
| должность, уч. Степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ОТЧЁТ О ЛАБОРАТОРНОЙ РАБОТЕ №5 | | | | | |
| пРИМЕНЕНИЕ ВЕРОЯТНОСТНО-СТАТИСТИЧЕСКИХ МЕТОДОВ ТЕОРИИ СИСТЕМ В ИНЖЕНЕРНО-НАУЧНЫХ ИССЛЕДОВАНИЯХ. АЛГОРИТМЫ МС И МАШИННОГО ОБУЧЕНИЯ | | | | | |
| по дисциплине: СИСТЕМНЫЙ АНАЛИЗ | | | | | |
|  | | | | | |
| РАБОТУ ВЫПОЛНИЛ | | | | | |
| СТУДЕНТ ГР. | 4330М |  |  |  | А.А. Кинько |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2023

**Цель работы**

Целью работы является освоение вероятностно-статистических методов теории систем в инженерно-научных исследованиях, алгоритмов мс и машинного обучения.

**Текст задания №1**

Согласно варианту №8:

Для приведенных исходных данных постройте диаграмму рассеяния и определите по ней характер зависимости. Рассчитайте выборочный коэффициент корреляции Пирсона, проверьте его значимость при α = 0.05. Запишите уравнение регрессии и дайте интерпретацию полученных результатов.

Исходные данные. Компанию по прокату автомобилей интересует зависимость между пробегом автомобиля (X) и стоимость ежемесячного технического обслуживания (Y). Для выяснения характера этой зависимости было отобрано 15 автомобилей (табл.).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|  | 13 | 16 | 15 | 20 | 19 | 21 | 26 | 24 | 30 | 32 | 30 | 35 | 34 | 40 | 39 |

**Текст задания №2**

Дана таблица с объектами.

Построим линейный классификатор, в котором выражение мажорируется функцией

|  |  |  |
| --- | --- | --- |
| Объекты | X | Y |
| A | -2 | -1 |
| B | -1 | -1 |
| C | 0 | 1 |
| D | 1 | 1 |
| E | 2 | 1 |

Получить следующее правило классификации: если то объект из класса 1,

иначе объект из класса -1 (ответ записать в виде десятичной дроби, округленной до

двух знаков после запятой).

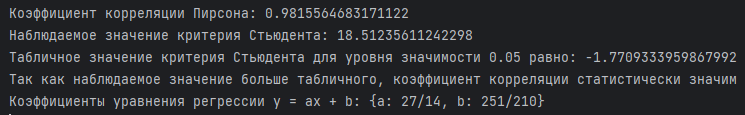
**Ход работы**

Для установления характера зависимости данных их задания №1 рассчитаем выборочный коэффициент корреляции Пирсона по формуле:

После этого проверим его значимость при . Для этого возьмем табличное значение и сравним его с экспериментальным значением. Если оно окажется больше, принимаем гипотезу о значимости критерия.

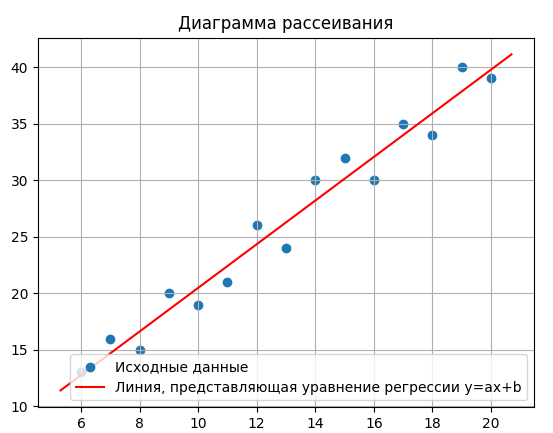
Построим уравнение регрессии , где

Для расчетов напишем скрипт Python, результат выполнения которого можно наблюдать ниже:



Так как коэффициент корреляции Пирсона , можно установить сильную связь данных.

На рисунке ниже представлена диаграмма рассеивания с нанесенной прямой уравнения регрессии.



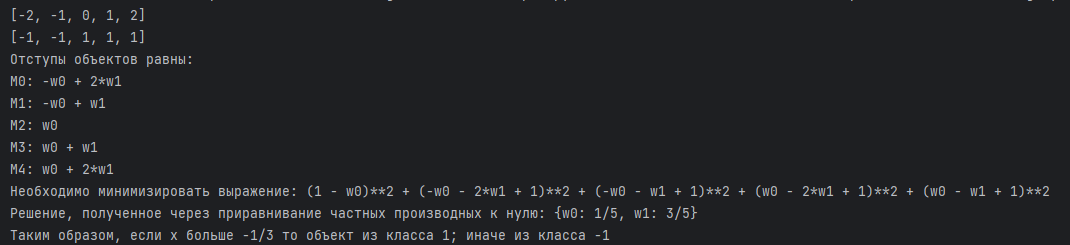
Для решения второй задачи сначала необходимо рассчитать отступы объектов по формуле:

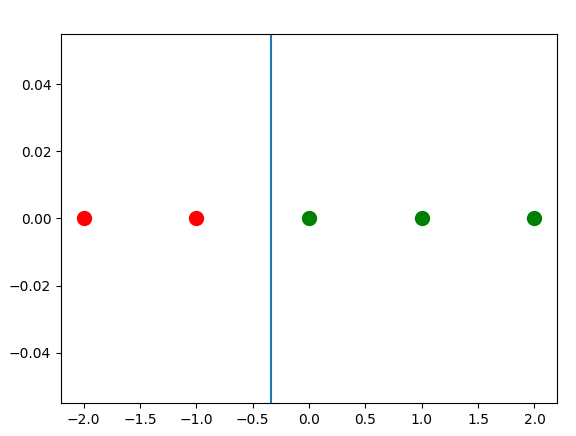
Затем минимизировать выражение через приравнивание частных производных к нулю:

После этого полученные коэффициенты преобразовать к пороговому значению:

Так, если , то объект будет относиться к классу 1; иначе – к классу -1.

Полученные с помощью скрипта Python результаты представлены ниже:





**Выводы**

В ходе выполнения лабораторной работы были получены теоретические основы вероятностно-статистических методов теории систем, алгоритмов мс и машинного обучения для решения практических задач. Так, для первой задачи были рассчитаны коэффициент корреляции, равный , который указал на сильную связь данных (проверен на значимость), а также уравнение регрессии: . Для второй задачи было найдено правило классификации: если , то объект принадлежит классу 1; иначе – классу -1.

**Приложение А**

Листинг скрипта Python для решения первой задачи.

import math  
import numpy as np  
from scipy.stats import t  
from sympy import symbols, Eq, solve  
  
import matplotlib.pyplot as plt  
  
  
def load\_data(path):  
 with open(path) as file:  
 x = [int(n) for n in file.readline().rsplit()]  
 y = [int(n) for n in file.readline().rsplit()]  
 return x, y  
  
  
def show\_scatter\_plot(x, y, coefs):  
 plt.scatter(x, y)  
 plt.grid(True)  
  
 axes = plt.gca()  
 x\_vals = np.array(axes.get\_xlim())  
 y\_vars = coefs.get(symbols('b')) + coefs.get(symbols('a')) \* x\_vals  
 plt.plot(x\_vals, y\_vars, color='red')  
 plt.legend(["Исходные данные", "Линия, представляющая уравнение регрессии y=ax+b"])  
 plt.title("Диаграмма рассеивания")  
 plt.show()  
  
  
def get\_correlation\_coefficient(x, y):  
 x2 = [pow(x\_i, 2) for x\_i in x]  
 y2 = [pow(y\_i, 2) for y\_i in y]  
 xy = [x[i] \* y[i] for i in range(len(x))]  
 up = sum(xy) - len(x) \* sum(x) / len(x) \* sum(y) / len(x)  
 down = math.sqrt(sum(x2) - len(x) \* pow(sum(x) / len(x), 2)) \* math.sqrt(sum(y2) - len(y) \* pow(sum(y) / len(y), 2))  
 return up / down  
  
  
def get\_student\_observ(c, n):  
 return c \* math.sqrt(n - 2) / math.sqrt(1 - pow(c, 2))  
  
  
def get\_student\_real(alpha, d):  
 return t.ppf(q=alpha, df=d)  
  
  
def find\_line(x, y):  
 a = symbols('a')  
 b = symbols('b')  
 eq1 = a \* sum([pow(x\_i, 2) for x\_i in x]) + b \* sum(x) - sum([x[i] \* y[i] for i in range(len(x))])  
 eq2 = a \* sum(x) + b \* len(x) - sum(y)  
 sol = solve([eq1, eq2], a, b)  
 return sol  
  
  
x\_data, y\_data = load\_data("LAB-5\_1\_Data.txt")  
corr = get\_correlation\_coefficient(x\_data, y\_data)  
print("Коэффициент корреляции Пирсона:", corr)  
s\_corr = get\_student\_observ(corr, len(x\_data))  
print("Наблюдаемое значение критерия Стьюдента:", s\_corr)  
  
alpha = 0.05  
s\_real = get\_student\_real(alpha, len(x\_data) - 2)  
print("Табличное значение критерия Стьюдента для уровня значимости", alpha, "равно:", s\_real)  
if abs(s\_corr) > abs(s\_real):  
 print("Так как наблюдаемое значение больше табличного, коэффициент корреляции статистически значим")  
  
line\_coefs = find\_line(x\_data, y\_data)  
print("Коэффициенты уравнения регрессии y = ax + b:", line\_coefs)  
  
show\_scatter\_plot(x\_data, y\_data, line\_coefs)

Файл «LAB-5\_1\_Data.txt»:

6 7 8 9 10 11 12 13 14 15 16 17 18 19 20  
13 16 15 20 19 21 26 24 30 32 30 35 34 40 39

Листинг скрипта Python для решения второй задачи.

# мажорирующая функция (1 - M)^2  
import matplotlib.pyplot as plt  
from sympy import symbols, Eq, solve, simplify, diff  
  
  
def load\_data(path):  
 with open(path) as file:  
 x = [int(n) for n in file.readline().rsplit()]  
 y = [int(n) for n in file.readline().rsplit()]  
 return x, y  
  
  
def calculate(x, y):  
 w1 = symbols("w1")  
 w0 = symbols("w0")  
 m = [y\_i \* (x\_i \* w1 + w0) for x\_i, y\_i in zip(x, y)]  
  
 print("Отступы объектов равны:")  
 for m\_no, m\_i in enumerate(m):  
 print("M" + str(m\_no) + ":", m\_i)  
  
 l = sum([pow(1 - m\_i, 2) for m\_i in m])  
 print("Необходимо минимизировать выражение:", l)  
 solution = solve([diff(l, w0), diff(l, w1)])  
 print("Решение, полученное через приравнивание частных производных к нулю:", solution)  
  
 x\_pass = -solution[w0] / solution[w1]  
 if solution[w1] > 0:  
 print("Таким образом, если x больше", x\_pass, "то объект из класса 1; иначе из класса -1")  
 else:  
 print("Таким образом, если x меньше", x\_pass, "то объект из класса 1; иначе из класса -1")  
  
 return x\_pass  
  
  
def show\_plot(x, y, x\_pass):  
 for i in range(len(x)):  
 if y[i] == 1:  
 plt.scatter([x[i]], [0], color='green', s=[100])  
 else:  
 plt.scatter([x[i]], [0], color='red', s=[100])  
  
 plt.axvline(x=x\_pass)  
 # plt.plot(x\_vals, y\_vars, color='red')  
 plt.show()  
  
  
  
x\_data, y\_data = load\_data("LAB-5\_2\_Data.txt")  
print(x\_data)  
print(y\_data)  
p = calculate(x\_data, y\_data)  
show\_plot(x\_data, y\_data, p)

Файл «LAB-5\_2\_Data.txt»:

-2 -1 0 1 2  
-1 -1 1 1 1