Міністерство освіти і науки України

Національний технічний університет України «КПІ» імені Ігоря Сікорського

Кафедра обчислювальної техніки ФІОТ

ЗВІТ

з лабораторної роботи №5

з навчальної дисципліни «Методи наукових досліджень»

Тема:

Проведення трьохфакторного експерименту при використанні рівняння регресії з урахуванням квадратичних членів

(центральний ортогональний композиційний план)

Виконала:

Студентка 2 курсу кафедри ОТ ФІОТ,

Навчальної групи ІВ-92

Орлова Ю.Д.

Номер у списку групи: 15

Перевірив:

Регіда П.Г.

Київ 2021

Мета: Провести трьохфакторний експеримент з урахуванням квадратичних членів, використовуючи центральний ортогональний композиційний план. Знайти рівняння регресії, яке буде адекватним для опису об'єкту.

Завдання на лабораторну роботу:

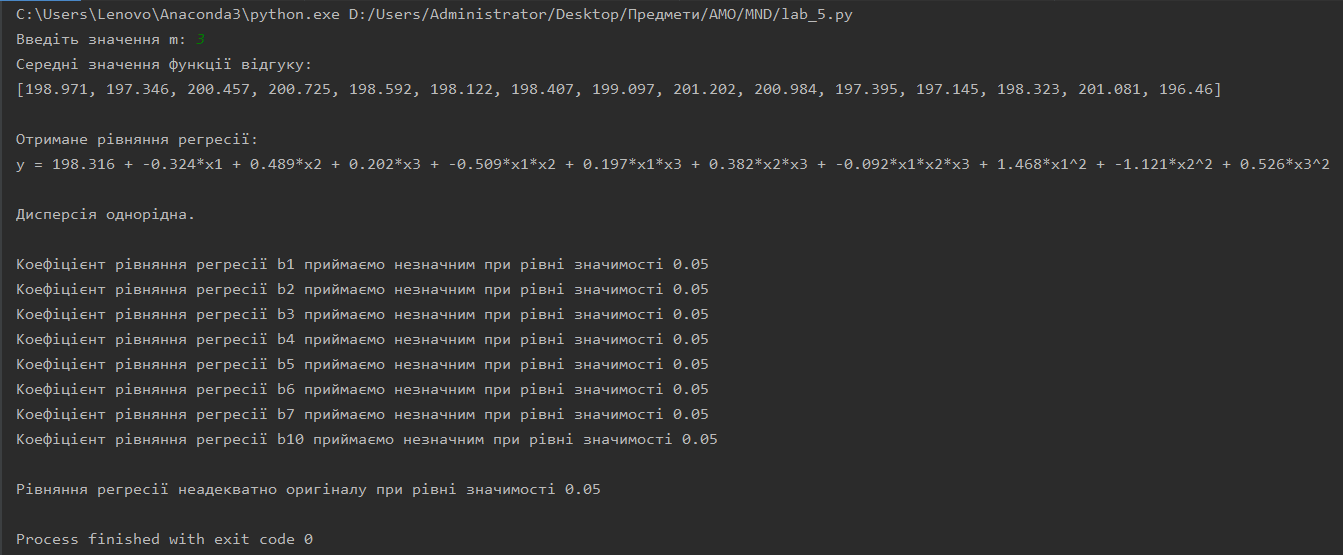
1. Взяти рівняння з урахуванням квадратичних членів.
2. Скласти матрицю планування для ОЦКП
3. Провести експеримент у всіх точках факторного простору (знайти значення функції відгуку Y). Значення функції відгуку знайти у відповідності з варіантом діапазону, зазначеного далі. Варіанти вибираються по номеру в списку в журналі викладача.
4. Розрахувати коефіцієнти рівняння регресії і записати його.
5. Провести 3 статистичні перевірки.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № варіанту | x1 | | x2 | | x3 | |
| 215 | min | max | min | max | min | max |
| -2 | 7 | -9 | 2 | -5 | 1 |

Код програми:

import random  
import numpy as np  
import scipy.stats  
from sklearn import linear\_model  
  
x1\_min = -2; x1\_max = 7  
x2\_min = -9; x2\_max = 2  
x3\_min = -5; x3\_max = 1  
  
y\_min = 200 + (x1\_min + x2\_min + x3\_min) / 3  
y\_max = 200 + (x1\_max + x2\_max + x3\_max) / 3  
  
l = 1.215  
n = 15  
  
x0\_n = (1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)  
x1\_n = (-1, -1, -1, -1, 1, 1, 1, 1, -l, l, 0, 0, 0, 0, 0)  
x2\_n = (-1, -1, 1, 1, -1, -1, 1, 1, 0, 0, -l, l, 0, 0, 0)  
x3\_n = (-1, 1, -1, 1, -1, 1, -1, 1, 0, 0, 0, 0, -l, l, 0)  
x1x2\_n = [x1\_n[i] \* x2\_n[i] for i in range(n)]  
x1x3\_n = [x1\_n[i] \* x3\_n[i] for i in range(n)]  
x2x3\_n = [x2\_n[i] \* x3\_n[i] for i in range(n)]  
x1x2x3\_n = [x1\_n[i] \* x2\_n[i] \* x3\_n[i] for i in range(n)]  
x1\_squared\_n = [x1\_n[i] \*\* 2 for i in range(n)]  
x2\_squared\_n = [x2\_n[i] \*\* 2 for i in range(n)]  
x3\_squared\_n = [x3\_n[i] \*\* 2 for i in range(n)]  
  
  
# values of factors for stellar points  
def value(x\_max, x\_min, l):  
 x0 = (x\_max + x\_min) / 2  
 delta\_x = x\_max - x0  
 return l \* delta\_x + x0  
  
  
x1 = (x1\_min, x1\_min, x1\_min, x1\_min, x1\_max, x1\_max, x1\_max, x1\_max, value(x1\_max, x1\_min, -l), value(x1\_max, x1\_min, l),  
 (x1\_max + x1\_min) / 2, (x1\_max + x1\_min) / 2, (x1\_max + x1\_min) / 2, (x1\_max + x1\_min) / 2, (x1\_max + x1\_min) / 2)  
x2 = (x2\_min, x2\_min, x2\_max, x2\_max, x2\_min, x2\_min, x2\_max, x2\_max, (x2\_max + x2\_min) / 2, (x2\_max + x2\_min) / 2,  
 value(x2\_max, x2\_min, -l), value(x2\_max, x2\_min, l), (x2\_max + x2\_min) / 2, (x2\_max + x2\_min) / 2, (x2\_max + x2\_min) / 2)  
x3 = (x3\_min, x3\_max, x3\_min, x3\_max, x3\_min, x3\_max, x3\_min, x3\_max, (x3\_max + x3\_min) / 2, (x3\_max + x3\_min) / 2,  
 (x3\_max + x3\_min) / 2, (x3\_max + x3\_min) / 2, value(x3\_max, x3\_min, -l), value(x3\_max, x3\_min, l), (x3\_max + x3\_min) / 2)  
  
x1x2 = [x1[i] \* x2[i] for i in range(n)]  
x1x3 = [x1[i] \* x3[i] for i in range(n)]  
x2x3 = [x2[i] \* x3[i] for i in range(n)]  
x1x2x3 = [x1[i] \* x2[i] \* x3[i] for i in range(n)]  
x1\_squared = [x1[i] \*\* 2 for i in range(n)]  
x2\_squared = [x2[i] \*\* 2 for i in range(n)]  
x3\_squared = [x3[i] \*\* 2 for i in range(n)]  
  
  
def experiment(m):  
 y = [[random.uniform(y\_min, y\_max) for i in range(m)] for j in range(n)]  
  
 # the average value of the response functions in the rows  
 y\_response = ([round(sum(y[j][i] for i in range(m)) / m, 3) for j in range(n)])  
  
 print('Середні значення функції відгуку:\n{0}'.format(y\_response))  
  
 b = list(zip(x0\_n, x1\_n, x2\_n, x3\_n, x1x2\_n, x1x3\_n, x2x3\_n, x1x2x3\_n, x1\_squared\_n, x2\_squared\_n, x3\_squared\_n))  
 skm = linear\_model.LinearRegression(fit\_intercept=False)  
 skm.fit(b, y\_response)  
 b = skm.coef\_  
 b = [round(i, 3) for i in b]  
  
 print('\nОтримане рівняння регресії:\ny = {0} + {1}\*x1 + {2}\*x2 + {3}\*x3 + {4}\*x1\*x2 + {5}\*x1\*x3 + {6}\*x2\*x3 + '  
 '{7}\*x1\*x2\*x3 + {8}\*x1^2 + {9}\*x2^2 + {10}\*x3^2\n'.format(round(b[0], 3), round(b[1], 3), round(b[2], 3),  
 round(b[3], 3), round(b[4], 3), round(b[5], 3), round(b[6], 3),  
 round(b[7], 3), round(b[8], 3), round(b[9], 3), round(b[10], 3)))  
  
 # checking the homogeneity of the variance according to the Cochren's criterion  
 dispersions = [sum([(y[j][i] - y\_response[j]) \*\* 2 for i in range(m)]) / m for j in range(n)]  
 gp = max(dispersions) / sum(dispersions)  
  
 f1 = m - 1; f2 = n; q = 0.05  
  
 if 11 <= f1 <= 16: f1 = 11  
 if 17 <= f1 <= 136: f1 = 17  
 if f1 > 136: f1 = 137  
 gt = {1: 0.9065, 2: 0.7679, 3: 0.6841, 4: 0.6287, 5: 0.5892, 6: 0.5598, 7: 0.5365, 8: 0.5365, 9: 0.5017, 10: 0.4884,  
 11: 0.4366, 17: 0.3720, 137: 0.2500}  
  
 if gp > gt[f1]:  
 i = input('Дисперсія неоднорідна. Якщо ви хочете повторити експериметн при m = m + 1 = {}, введіть 1: \n'  
 .format(m + 1))  
 if i == '1':  
 experiment(m + 1)  
 m += 1  
 else:  
 print('Дисперсія однорідна.\n')  
  
 # assessment of the significance of regression coefficients according to Student's criterion  
 s\_b = sum(dispersions) / n  
 s = np.sqrt(s\_b / (n \* m))  
  
 t = [abs(b[i]) / s for i in range(11)]  
  
 f3 = f1 \* f2  
  
 d = 0  
 for i in range(11):  
 if t[i] < scipy.stats.t.ppf(q=0.975, df=f3):  
 print('Коефіцієнт рівняння регресії b{0} приймаємо незначним при рівні значимості 0.05'.format(i))  
 b[i] = 0  
 else:  
 d += 1  
  
 # Fisher's criterion  
 f4 = n - d  
  
 s\_ad = (m \* sum([(b[0] + b[1] \* x1\_n[i] + b[2] \* x2\_n[i] + b[3] \* x3\_n[i] + b[4] \* x1\_n[i] \* x2\_n[i] + b[5] \*  
 x1\_n[i] \* x3\_n[i] + b[6] \* x2\_n[i] \* x3\_n[i] + b[7] \* x1\_n[i] \* x2\_n[i] \* x3\_n[i] -  
 y\_response[i]) \*\* 2 for i in range(n)]) / f4)  
 f\_p = s\_ad / s\_b  
  
 if f\_p > scipy.stats.f.ppf(q=0.95, dfn=f4, dfd=f3):  
 print(' \nРівняння регресії неадекватно оригіналу при рівні значимості 0.05')  
 else:  
 print(' \nРівняння регресії адекватно оригіналу при рівні значимості 0.05')  
  
  
try:  
 m = int(input(("Введіть значення m: ")))  
 experiment(m)  
except:  
 breakpoint()  
 print("Ви ввели не ціле число. Спробуйте знову.")

Результат виконання програми:



Висновки: під час виконання програми ми провели трьохфакторний експеримент з урахуванням квадратичних членів, використовуючи центральний ортогональний композиційний планта знайшли рівняння регресії, яке буде адекватним для опису об'єкту. Було написано програму з використанням можливостей алгоритмічної мови високого рівня Python, яка це все виконує. Результати роботи програми підтвердили правильність її виконання.