Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра автоматики та управління в технічних системах

Лабораторна робота №4

Проведення трьохфакторного експерименту при використанні рівняння регресії з урахуванням ефекту взаємодії.

Виконала

студентка групи ІТ-91: Перевірила:

Луцай Катерина Сокульський О. Є.

Київ 2022

Мета: Провести повний трьохфакторний експеримент. Знайти рівняння регресії адекватне об'єкту.

Завдання

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.

$$y_{i\max} = 200 + x_{cp\max}$$
$$y_{i\min} = 200 + x_{cp\min}$$

где $x_{cp\max} = \frac{x_{1\max} + x_{2\max} + x_{3\max}}{3}$, $x_{cp\min} = \frac{x_{1\min} + x_{2\min} + x_{3\min}}{3}$

4. Розрахувати коефіцієнти рівняння регресії і записати його.
5. Провести 3 статистичні перевірки.

Варіант 15

```
import numpy as np
from sklearn.preprocessing import minmax_scale

x1_min, x1_max = -25, 75
x2_min, x2_max = 25, 65
x3_min, x3_max = 25, 40

y_min = 200 + (x1_min + x2_min + x3_min)/3
y_max = 200 + (x1_max + x2_max + x3_max)/3

n = 8  # кількість точок
c = 8  # кількість коефіцієнтів

norm_factors = np.array([[1, 1, 1],
                         [1, 1, -1],
                         [1, -1, -1],
                         [-1, -1, -1],
                         [-1, 1, 1],
                         [1, -1, 1],
                         [-1, -1, 1],
                         [-1, 1, -1]])

# інтеракції між факторами
def interact_factors(factors):
    full_factors = np.zeros((n, 4))
    full_factors[:, 0] = factors[:, 0] * factors[:, 1]
    full_factors[:, 1] = factors[:, 0] * factors[:, 2]
    full_factors[:, 2] = factors[:, 1] * factors[:, 2]
    full_factors[:, 3] = factors[:, 0] * factors[:, 1] * factors[:, 2]
    return full_factors

# отримання натуральних факторів з нормованих
def natural(norm_factors):
    natural = np.zeros_like(norm_factors)
    natural[:, 0][norm_factors[:, 0] == 1] = x1_max
    natural[:, 1][norm_factors[:, 1] == 1] = x2_max
    natural[:, 2][norm_factors[:, 2] == 1] = x3_max
```

```
    natural[:, 0][norm_factors[:, 0] == -1] = x1_min
    natural[:, 1][norm_factors[:, 1] == -1] = x2_min
    natural[:, 2][norm_factors[:, 2] == -1] = x3_min
    return natural

# регресія за факторами та коефцієнтами
def regression(matrix, coefs, n, c):
    def func(factors, coefs, c):
        y = coefs[0]
        for i in range(c):
            y += coefs[i+1] * factors[i]
        return y
    values = np.zeros(n)
    for f in range(n):
        values[f] = func(matrix[f], coefs, c)
    return values

# вирішення нормованого СЛР
def solve_norm_coef(factors, values):
    coefs = np.zeros(c)
    coefs[0] = values.mean()
    for i in range(c-1):
        coefs[i+1] = np.mean(values * factors[:, 0])
    return coefs

# вирішення натурального СЛР
def solve_coef(factors, values):
    A = np.zeros((c, c))
    x_mean = np.mean(factors, axis=0)
    for i in range(1, c):
        for j in range(1, c):
            A[j, i] = np.mean(factors[:, i-1] * factors[:, j-1])
    A[0, 0] = n
    A[0, 1:] = x_mean
    A[1:, 0] = x_mean
    B = np.mean(np.tile(values, (c, 1)).T * np.column_stack((np.ones(n),
factors)), axis=0)
    coef = np.linalg.solve(A, B)
    return coef

print(f"X1 min: {x1_min}\tX1 max: {x1_max}")
print(f"X2 min: {x2_min}\tX2 max: {x2_max}")
print(f"X3 min: {x3_min}\tX3 max: {x3_max}")
print(f"Y min: {y_min}\tY max: {y_max}")

# отримання оптимальної кількості випробувань для однієї комбінації факторів
Gt = [6.798, 5.157, 4.737, 3.91]
for i in range(2, 10):
    rand_y = np.random.randint(y_min, y_max, (n, i))
    std_y = np.std(rand_y, axis=1)**2
    print(f"Max Y dispersion: {std_y.max()}")
    Gp = std_y.max()/std_y.mean()
    f1 = i - 1
    print(f"m = {i} Gp: {Gp}\tGt: {Gt[i-2]}")
    if Gp < Gt[i-2]:
        m = i  # кількість випробувань
        break

print(f"Mean Y dispersion: {std_y.mean()}")
```

```
# формування факторів
factors = natural(norm_factors)
factors = np.concatenate((factors, interact_factors(factors)), axis=1)
norm_factors = np.concatenate((norm_factors, interact_factors(norm_factors)),
axis=1)

# генерація Y значень для експериментів
rand_y = np.random.randint(y_min, y_max, (n, m))
norm_y = minmax_scale(rand_y.reshape(n*m), feature_range=(-1,1)).reshape(n ,m)

# перевірка за критеріїм стюдента
coefs_value = np.zeros(4)
coefs_value[0] = np.mean(rand_y.mean(axis=1))
for i in range(3):
    coefs_value[i+1] = np.mean(rand_y.mean(axis=1) * factors[:, i])
stud_crit = np.abs(coefs_value) / np.sqrt(std_y.mean()/(n*m))
ts = [2.306, 2.12, 2.064]
sig_coefs = len(stud_crit[stud_crit > ts[m-2]])
print(f"All coefs are significant: {sig_coefs == m}\t{sig_coefs}")

# формування планів
norm_plan = np.concatenate((norm_factors, norm_y), axis=1)
norm_plan = np.concatenate((norm_plan, norm_y.mean(axis=1).reshape(n, 1)),
axis=1)
std_norm_y = np.std(norm_y, axis=1)**2
norm_plan = np.concatenate((norm_plan, std_norm_y.reshape(n, 1)), axis=1)
print(f"Normalized plan:\n{norm_plan}")
plan = np.concatenate((factors, rand_y), axis=1)
plan = np.concatenate((plan, rand_y.mean(axis=1).reshape(n, 1)), axis=1)
plan = np.concatenate((plan, std_y.reshape(n, 1)), axis=1)
print(f"Natural plan:\n{plan}")

norm_coefs = solve_norm_coef(norm_factors, norm_y.mean(axis=1))
print("Normalized coefs", norm_coefs)
coefs = solve_coef(factors, rand_y.mean(axis=1))
print("Natural coefs", coefs)

reg_val = regression(norm_factors, norm_coefs, n, m)
print("Norm mean Y - regression Y")
print(np.column_stack((norm_y.mean(axis=1), reg_val)))

# перевірка критерію Фішера
f3 = n * (m -1) # 8
f4 = n - sig_coefs # 4
disp = m/(n - sig_coefs) * np.sum((reg_val - norm_y.mean(axis=1))**2)
Fp = disp / std_y.mean()
Ft = 3.8
print(f"Fisher crit: {Fp < Ft}\t{Fp}\t{Ft}")
```

Результат виконання:

```
X1 min: -25     X1 max: 75

X2 min: 25 X2 max: 65
```

X3 min: 25 X3 max: 40

Y min: 208.33333333333334 Y max: 260.0

Max Y dispersion: 342.25

m = 2 Gp: 2.535185185185185 Gt: 6.798

Mean Y dispersion: 135.0

All coefs are significant: False 4

Normalized plan:

```
[[ 1.          1.          1.          1.          1.          1.
   1.         -1.          0.76744186 -0.11627907  0.78096268]
 [ 1.          1.         -1.          1.         -1.         -1.
  -1.         -0.02325581  0.20930233  0.09302326  0.01352082]
 [ 1.         -1.         -1.         -1.         -1.          1.
   1.         -0.86046512 -0.44186047 -0.65116279  0.04380746]
 [-1.         -1.         -1.          1.          1.          1.
  -1.          1.         -1.          0.          1.        ]
 [-1.          1.          1.         -1.         -1.          1.
  -1.          0.44186047  0.90697674  0.6744186   0.05408329]
 [ 1.         -1.          1.         -1.          1.         -1.
  -1.          0.53488372  0.86046512  0.69767442  0.02650081]
 [-1.         -1.          1.          1.         -1.         -1.
   1.         -0.06976744  0.11627907  0.02325581  0.00865333]
 [-1.          1.         -1.         -1.          1.         -1.
   1.          0.34883721 -0.1627907   0.09302326  0.06544078]]
```

Natural plan:

```
[[ 7.50000e+01  6.50000e+01  4.00000e+01  4.87500e+03  3.00000e+03
   2.60000e+03  1.95000e+05  2.15000e+02  2.53000e+02  2.34000e+02
```

```
   2.40250e+02]
 [ 7.50000e+01  6.50000e+01  2.50000e+01  4.87500e+03  1.87500e+03
   1.62500e+03  1.21875e+05  2.36000e+02  2.41000e+02  2.38500e+02
   4.00000e+00]
 [ 7.50000e+01  2.50000e+01  2.50000e+01  1.87500e+03  1.87500e+03
   6.25000e+02  4.68750e+04  2.18000e+02  2.27000e+02  2.22500e+02
   4.90000e+01]
 [-2.50000e+01  2.50000e+01  2.50000e+01 -6.25000e+02 -6.25000e+02
   6.25000e+02 -1.56250e+04  2.58000e+02  2.15000e+02  2.36500e+02
   1.44000e+02]
 [-2.50000e+01  6.50000e+01  4.00000e+01 -1.62500e+03 -1.00000e+03
   2.60000e+03 -6.50000e+04  2.46000e+02  2.56000e+02  2.51000e+02
   3.42250e+02]
 [ 7.50000e+01  2.50000e+01  4.00000e+01  1.87500e+03  3.00000e+03
   1.00000e+03  7.50000e+04  2.48000e+02  2.55000e+02  2.51500e+02
   2.56000e+02]
 [-2.50000e+01  2.50000e+01  4.00000e+01 -6.25000e+02 -1.00000e+03
   1.00000e+03 -2.50000e+04  2.35000e+02  2.39000e+02  2.37000e+02
   2.25000e+00]
 [-2.50000e+01  6.50000e+01  2.50000e+01 -1.62500e+03 -6.25000e+02
   1.62500e+03 -4.06250e+04  2.44000e+02  2.33000e+02  2.38500e+02
   4.22500e+01]]
Normalized coefs [ 0.10174419 -0.09593023 -0.09593023 -0.09593023 -
0.09593023 -0.09593023
 -0.09593023 -0.09593023]
Natural coefs [ 2.07118366e-01  5.61926386e-01  4.14379402e+00
6.83032576e+00
```

```
 -9.80035219e-03 -1.28259394e-02 -1.16756904e-01  1.84055232e-04]
Norm mean Y - regression Y
[[-0.11627907 -0.09011628]
 [ 0.09302326 -0.09011628]
 [-0.65116279  0.10174419]
 [ 0.          0.29360465]
 [ 0.6744186   0.10174419]
 [ 0.69767442  0.10174419]
 [ 0.02325581  0.29360465]
 [ 0.09302326  0.10174419]]
Fisher crit: True     0.005346483584720466  3.8
```

Висновки: було проведено проведено повний трьохфакторний експеримент, знайдено рівняння регресії адекватне об'єкту.