# importing the required packages and libraries

from scipy.optimize import curve\_fit

from numpy import array

import matplotlib.pyplot as plt

# defining the variables

values\_x = array([0.079, 0.431, 0.782, 1.134, 1.486, 1.838, 2.189, 2.541, 2.893])

values\_y = array([-4.308, -1.778, -0.268, 1.144, 2.248, 3.273, 4.440, 5.396, 6.357])

# defining objective functions

def mapping1(values\_x, a, b, c):

    return a \* values\_x\*\*2 + b \* values\_x + c

def mapping2(values\_x, a, b, c):

    return b \* values\_x\*\*a + c

# using the curve\_fit() function

args, \_ = curve\_fit(mapping1, values\_x, values\_y)

a, b, c = args[0], args[1], args[2]

y\_fit1 = a \* values\_x\*\*2 + b \* values\_x + c

print()

print("y = a \* x^2 + b \* x + c")

print("a =", a, "  b =", b, "  c =", c)

args, \_ = curve\_fit(mapping2, values\_x, values\_y)

a, b, c = args[0], args[1], args[2]

y\_fit2 = b \* values\_x\*\*a + c

print()

print("y = b \* x^a + c")

print("a =", a, "  b =", b, "  c =", c)

# plotting the graph

plt.plot(values\_x, values\_y, "bo", label="y - original")

plt.plot(values\_x, y\_fit1, label="y = a \* x^2 + b \* x + c")

plt.plot(values\_x, y\_fit2, label="y = b \* x^a + c")

plt.xlabel("x")

plt.ylabel("y")

plt.legend(loc="best", fancybox=True, shadow=True)

plt.grid(True)

plt.show()