```python
import numpy as np

indata = np.loadtxt(r"C:\Users\Zhenc_000\Documents\Caltech\CS156\Set7\indta.txt")
outdata = np.loadtxt(r"C:\Users\Zhenc_000\Documents\Caltech\CS156\Set7\outdta.txt")

def linreg(x,y): #function to perform linear regression
    return np.dot(np.linalg.pinv(x),y)

def error(w,z,y): #find the in-sample or out of sample error
    g = np.sign(np.dot(z,np.transpose(w)))
    return ((len(y)-np.sum(np.dot(y,g)))/2)/len(y)

in_x1 = indata[:24,0]
in_x2 = indata[:24,1]
in_y = indata[:24,2]
in_x1_val = indata[25:,0]
in_x2_val = indata[25:,1]
in_y_val = indata[25:,2]
out_x1 = outdata[:,0]
out_x2 = outdata[:,1]
out_y = outdata[:,2]

in_z = np.zeros([len(in_x1),8]) #transform x to z
in_z[:,0] = 1
in_z[:,1] = in_x1
in_z[:,2] = in_x2
in_z[:,3] = np.multiply(in_x1,in_x1)
in_z[:,4] = np.multiply(in_x2,in_x2)
in_z[:,5] = np.multiply(in_x1,in_x2)
in_z[:,6] = np.absolute(in_x1-in_x2)
in_z[:,7] = np.absolute(in_x1+in_x2)

in_z_val = np.zeros([len(in_x1_val),8]) #transform x to z
in_z_val[:,0] = 1
in_z_val[:,1] = in_x1_val
in_z_val[:,2] = in_x2_val
in_z_val[:,3] = np.multiply(in_x1_val,in_x1_val)
in_z_val[:,4] = np.multiply(in_x2_val,in_x2_val)
in_z_val[:,5] = np.multiply(in_x1_val,in_x2_val)
in_z_val[:,6] = np.absolute(in_x1_val-in_x2_val)
in_z_val[:,7] = np.absolute(in_x1_val+in_x2_val)

out_z = np.zeros([len(out_x1),8]) #transform x to z
out_z[:,0] = 1
out_z[:,1] = out_x1
out_z[:,2] = out_x2
out_z[:,3] = np.multiply(out_x1,out_x1)
out_z[:,4] = np.multiply(out_x2,out_x2)
out_z[:,5] = np.multiply(out_x1,out_x2)
out_z[:,6] = np.absolute(out_x1-out_x2)
out_z[:,7] = np.absolute(out_x1+out_x2)

for i in range(4,9):
    w = linreg(in_z[:,0:i],in_y)
    w2 = linreg(in_z_val[:,0:i],in_y_val)
    print(error(w,in_z_val[:,0:i],in_y_val))
    print(error(w,out_z[:,0:i],out_y))
    print(error(w2,in_z[:,0:i],in_y))
    print(error(w2,out_z[:,0:i],out_y))
```