```python
import numpy as np
import random
from math import pow

def transform_poly(x_data, Q, output):
    for i in range(len(x_data)):
        output.append([])
        data_len = len(x_data[i])
        output[i].append(1.0)

        for j in range(Q):
            for k in range(data_len):
                output[i].append(pow(x_data[i][k],j+1))

def transform_fullorder(x_data, output):
    for i in range(len(x_data)):
        output.append([])
        data_len = len(x_data[i])
        output[i].append(1.0)

        for j in range(data_len):
                output[i].append(x_data[i][j])

        for j in range(data_len):
            for k in range(data_len)[j:]:
                output[i].append(x_data[i][j]*x_data[i][k])


def transform_lower(x_data, n, output):
    for i in range(len(x_data)):
        output.append([])
        output[i].append(1.0)

        for j in range(n):
            output[i].append(x_data[i][j])

def transform_random(x_data, output):
    for i in range(len(x_data)):
        output.append([])
        sample = []
        for j in range(len(x_data[i])):
            sample.append(j)

        data_len = len(x_data[i])
        output[i].append(1.0)
        sample = random.sample(sample, 5)

        for j in range(5):
            output[i].append(x_data[i][sample[j]])

def main():
    iter = 1                                #第十六題改200
    train_x = []
    train_y = []
    test_x = []
    test_y = []
    e_in = 0.0
    e_out = 0.0
    train_size = 0
    test_size = 0
    lower = 8

    with open('hw3_train.dat.txt', 'r') as f:
        while True:
            line = f.readline()
            if line == '\n' or len(line) == 0:
```

```
            break
        x = [float(i) for i in line.split()[:-1]]
        train_x.append(x)
        train_y.append([float(line.split()[-1])])
        train_size+=1

    with open('hw3_test.dat.txt', 'r') as f:
        while True:
            line = f.readline()
            if line == '\n' or len(line) == 0:
                break
            x = [float(i) for i in line.split()[:-1]]
            test_x.append(x)
            test_y.append([float(line.split()[-1])])
            test_size+=1


    for i in range(iter):
        random.seed(i)
        train_X = []
        train_Y = train_y.copy()
        test_X = []
        test_Y = test_y.copy()

        transform_poly(train_x, 8, train_X)        #12,13題用這個改Q的數值
        transform_fullorder(train_x, train_X)      #14題用這個
        transform_lower(train_x, lower, train_X)   #15題用這個
        transform_random(train_x, train_X)         #16題用這個

        train_X = np.array(train_X)
        train_Y = np.array(train_Y)

        w_op = np.linalg.pinv(train_X).dot(train_Y)
        predict_trainy = train_X.dot(w_op)

        e_in+=((np.sign(predict_trainy)!=train_Y).sum()/train_size)

        transform_poly(test_x, 8, test_X)          #12,13題用這個改Q的數值
        transform_fullorder(test_x, test_X)        #14題用這個
        transform_lower(test_x, lower, test_X)     #15題用這個
        transform_random(test_x, test_X)           #16題用這個

        test_X = np.array(test_X)
        test_Y = np.array(test_Y)

        predict_testy = test_X.dot(w_op)

        e_out+=((np.sign(predict_testy)!=test_Y).sum()/test_size)

    print((e_out-e_in)/iter)


if __name__ == '__main__':
    main()
```