```
# for questions #1-5

import numpy as np

def importData(filename):
    data = open(filename, 'r')
    lines = data.read().splitlines()

    matx = np.zeros(shape=(len(lines),3))
    vecty = np.zeros(shape=(len(lines),1))

    for i in range(len(lines)):
        point = lines[i].strip(' ').split('  ')
        matx[i] = np.matrix([1, float(point[0]), float(point[1])])
        vecty[i] = float(point[2])

    return matx, vecty


def transform(dataset, k):
    matx, vecty = dataset
    matz = np.zeros(shape=(len(matx), k+1))
    for i in range(len(matx)):
        x1 = matx[i, 1]
        x2 = matx[i, 2]

        z = np.matrix([1, x1, x2, x1 ** 2, x2 ** 2, x1 * x2,
            abs(x1 - x2), abs(x1 + x2)])
        matz[i] = z[0, :k+1]

    return matz


def calculateError(w, dataset):
    matz, vecty = dataset
    errorCount = 0
    for i in range(len(matz)):
        if np.sign(w.transpose().dot(matz[i, :])) != np.sign(vecty[i]):
            errorCount += 1

    return errorCount / len(matz)


#1-2
def validation(k):
    dataset = importData('in.dta.txt')
    matx, vecty = dataset
    matz = transform(dataset, k)
    trainMatz = matz[:25]
    testMatz = matz[25:35]

    #find w
    w = np.linalg.pinv(trainMatz).dot(vecty[:25])

    #calculate E_val
    valError = calculateError(w, (testMatz, vecty[25:35]))

    #calculate E_out
    outData = importData('out.dta.txt')
    outMatz = transform(outData, k)
```

```
    outError = calculateError(w, (outMatz, outData[1]))

    print(valError, outError)


#3-4
def validation2(k):
    dataset = importData('in.dta.txt')
    matx, vecty = dataset
    matz = transform(dataset, k)
    trainMatz = matz[25:35]
    testMatz = matz[:25]

    #find w
    w = np.linalg.pinv(trainMatz).dot(vecty[25:35])

    #calculate E_val
    valError = calculateError(w, (testMatz, vecty[:25]))

    #calculate E_out
    outData = importData('out.dta.txt')
    outMatz = transform(outData, k)
    outError = calculateError(w, (outMatz, outData[1]))

    print(valError, outError)


#1-2
for i in range(3, 8):
    validation(i)

#3-4
for i in range(3, 8):
    validation2(i)
```