

```

import numpy as np
import random

misc = 0
n = 100
wnew = np.zeros([6])

for t in range(1000):

    points = np.zeros([n, 6])
    sample = np.zeros([1000, 6])
    y = np.zeros([n])
    ytrue = np.zeros([n])
    ysample = np.zeros([1000])

    for i in range(n): #same algorithm as before except with xy, x2, and y2 terms
        points[i,0] = 1 #set all x0 to 1
        points[i,1] = np.random.uniform(-1,1) #set all x1 to the x-coordinate of point
        points[i,2] = np.random.uniform(-1,1) #set all x2 to the y-coordinate of point
        points[i,3] = np.dot(points[i,1],points[i,2])
        points[i,4] = np.square(points[i,1])
        points[i,5] = np.square(points[i,2])
        ytrue[i] = np.sign(np.square(points[i,1]) + np.square(points[i,2]) - 0.6)
        if np.random.randint(0,10) != 1: #randomly choose points that are "noisy"
            y[i] = np.sign(np.square(points[i,1]) + np.square(points[i,2]) - 0.6)
        else:
            y[i] = -1*np.sign(np.square(points[i,1]) + np.square(points[i,2]) - 0.6)

    for i in range(1000): #set up the sampling data
        sample[i,0] = 1
        sample[i,1] = np.random.uniform(-1,1)
        sample[i,2] = np.random.uniform(-1,1)
        sample[i,3] = np.dot(sample[i,1],sample[i,2])
        sample[i,4] = np.square(sample[i,1])
        sample[i,5] = np.square(sample[i,2])
        ysample[i] = np.sign(np.square(sample[i,1]) + np.square(sample[i,2]) - 0.6)

    w =
    np.dot(np.dot(np.linalg.inv(np.dot(np.matrix.transpose(points),points)),np.matrix.transpose(p
oints)),y)
    wnew += w #add up the w guesses to average later
    g = np.sign(np.dot(sample,w))
    misc += (1000-np.sum(np.dot(g,ysample)))/2

print(misc/1000000)
print(wnew/1000)

```