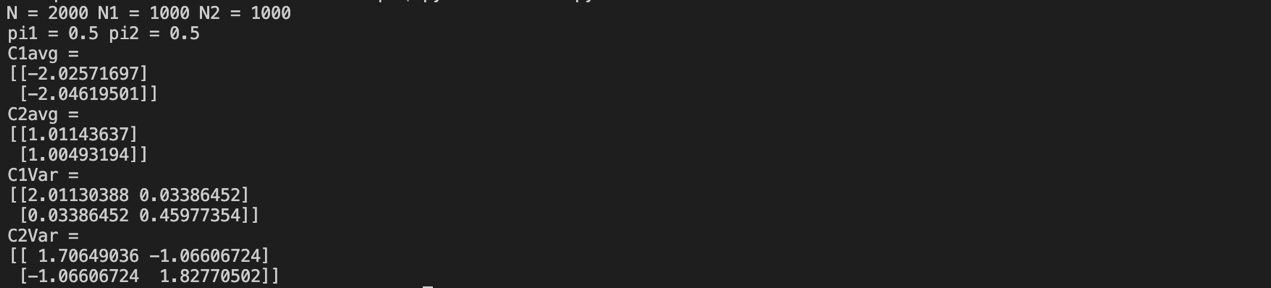
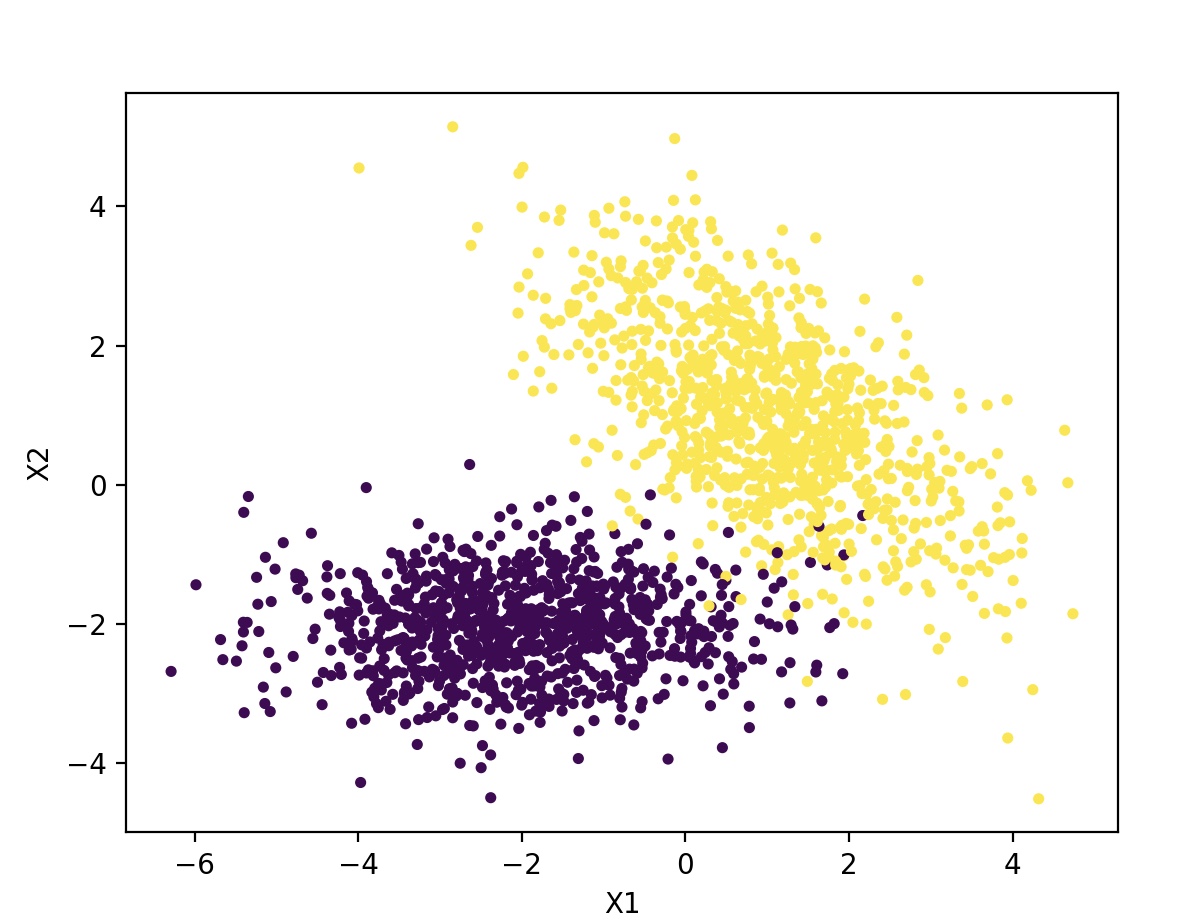
3.

(a)



(b)

(C)



我認為(b)的表現會比較好，因為Class 1和Class2的分佈走向顯示兩個種類的covariance matrix並不相同，Class 1的兩變數covariance應接近於零，但Class 兩變數應有一定程度相關，故兩者的covariance matrix不應假設相同。

Code:

import numpy as np

import matplotlib.pyplot as plt

# datashape = (2000, 3)

data = np.load("./data.npy")

# class count

N = 0

N1 = 0

N2 = 0

for i in range(2000):

N += 1

if data[i, 2] == 0:

N1 += 1

else:

N2 += 1

print("N = " + str(N), "N1 = " + str(N1), "N2 = " + str(N2))

print("pi1 = " + str(N1 / N), "pi2 = " + str(N2 / N))

# calculate average

X1 = np.array([data[:, 0] \* (1 - data[:, 2]), data[:, 1] \* (1 - data[:, 2])])

X2 = np.array([data[:, 0] \* data[:, 2], data[:, 1] \* data[:, 2]])

C1avg = (np.sum(X1, axis = 1) / N1).reshape(2, 1)

C2avg = (np.sum(X2, axis = 1) / N2).reshape(2, 1)

print("C1avg = ")

print(C1avg)

print("C2avg = ")

print(C2avg)

# calculate variance

X1 = np.array([(data[:, 0] - C1avg[0])\* (1 - data[:, 2]), (data[:, 1] - C1avg[1]) \* (1 - data[:, 2])])

X2 = np.array([(data[:, 0] - C2avg[0])\* data[:, 2], (data[:, 1] - C2avg[1]) \* data[:, 2]])

C1Var = np.matmul(X1, np.transpose(X1)) / N1

C2Var = np.matmul(X2, np.transpose(X2)) / N2

'''

print("C1Var = ")

print(C1Var)

print("C2Var = ")

print(C2Var)

'''

CVar = N1 / N \* C1Var + N2 / N \* C2Var

print("Cvar = ")

print(CVar)

# plot data

plt.scatter(data[:, 0], data[:, 1], marker = '.', c = data[:, 2])

plt.xlabel("X1")

plt.ylabel("X2")

plt.show()