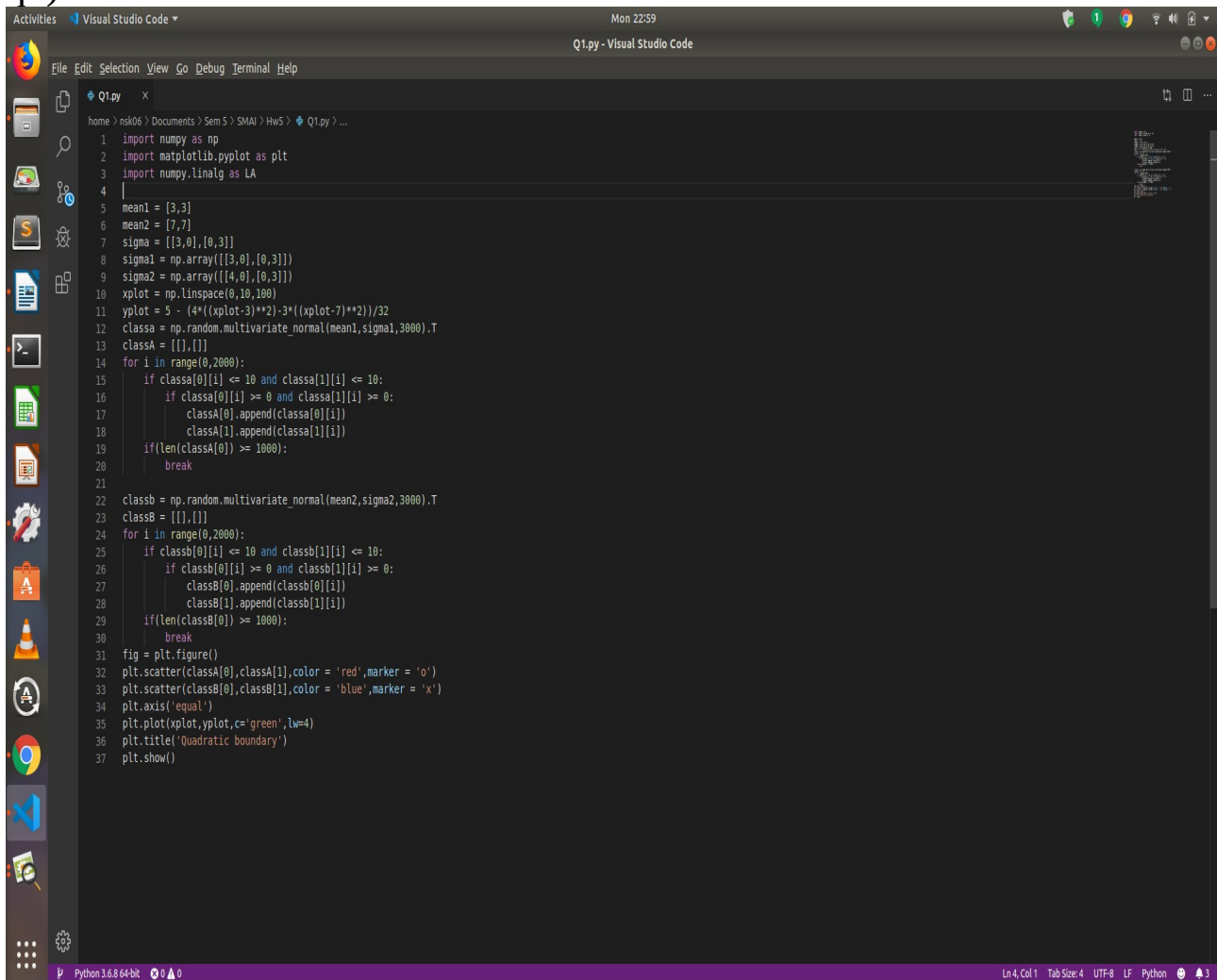


# Homework – 5

q1)



```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import numpy.linalg as LA
4
5 mean1 = [3,3]
6 mean2 = [7,7]
7 sigma = [[3,0],[0,3]]
8 sigma1 = np.array([[3,0],[0,3]])
9 sigma2 = np.array([[4,0],[0,3]])
10 xplot = np.linspace(0,10,100)
11 yplot = 5 - (4*((xplot-3)**2)-3*((xplot-7)**2))/32
12 classa = np.random.multivariate_normal(mean1,sigma1,3000).T
13 classA = [[],[]]
14 for i in range(0,2000):
15     if classa[0][i] <= 10 and classa[1][i] <= 10:
16         if classa[0][i] >= 0 and classa[1][i] >= 0:
17             classA[0].append(classa[0][i])
18             classA[1].append(classa[1][i])
19         if(len(classA[0]) >= 1000):
20             break
21
22 classb = np.random.multivariate_normal(mean2,sigma2,3000).T
23 classB = [[],[]]
24 for i in range(0,2000):
25     if classb[0][i] <= 10 and classb[1][i] <= 10:
26         if classb[0][i] >= 0 and classb[1][i] >= 0:
27             classB[0].append(classb[0][i])
28             classB[1].append(classb[1][i])
29         if(len(classB[0]) >= 1000):
30             break
31
32 fig = plt.figure()
33 plt.scatter(classA[0],classA[1],color = 'red',marker = 'o')
34 plt.scatter(classB[0],classB[1],color = 'blue',marker = 'x')
35 plt.plot(xplot,yplot,c='green',lw=4)
36 plt.title('Quadratic boundary')
37 plt.show()
```

This is the code to generate the plots the analytical equations are:

Ans = 1 a)  $y_1 = \begin{bmatrix} 3 \\ 3 \end{bmatrix}$   $y_2 = \begin{bmatrix} 7 \\ 7 \end{bmatrix}$

$$\Sigma_1 = \Sigma_2 = \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix}$$

The decision boundary here is

$$[X - y_1]^T \Sigma^{-1} [X - y_1] = [X - y_2]^T \Sigma^{-1} [X - y_2]$$

$$\begin{bmatrix} x-3 \\ y-3 \end{bmatrix}^T \frac{1}{9} \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} x-3 \\ y-3 \end{bmatrix} = \begin{bmatrix} x-7 \\ y-7 \end{bmatrix}^T \frac{1}{9} \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} x-7 \\ y-7 \end{bmatrix}$$

$$\Rightarrow (x-3)^2 + (y-3)^2 = (x-7)^2 + (y-7)^2$$

$$\Rightarrow \boxed{y = 10 - x}$$

b)  $\Sigma_1 = \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix}$   $\Sigma_2 = \begin{bmatrix} 4 & 0 \\ 0 & 3 \end{bmatrix}$

$$\begin{bmatrix} x-3, y-3 \end{bmatrix} \frac{1}{9} \begin{bmatrix} 3(x-3) \\ 3(y-3) \end{bmatrix} = \begin{bmatrix} x-7, y-7 \end{bmatrix} \frac{1}{12} \begin{bmatrix} 3 & 0 \\ 0 & 4 \end{bmatrix} \begin{bmatrix} x-7 \\ y-7 \end{bmatrix}$$

$$\frac{1}{9} (3(x-3)^2 + 3(y-3)^2) = \frac{1}{12} (3(x-7)^2 + 4(y-7)^2)$$

$$\frac{(x-3)^2}{3} + \frac{(y-3)^2}{3} = \frac{(x-7)^2}{4} + \frac{(y-7)^2}{3}$$

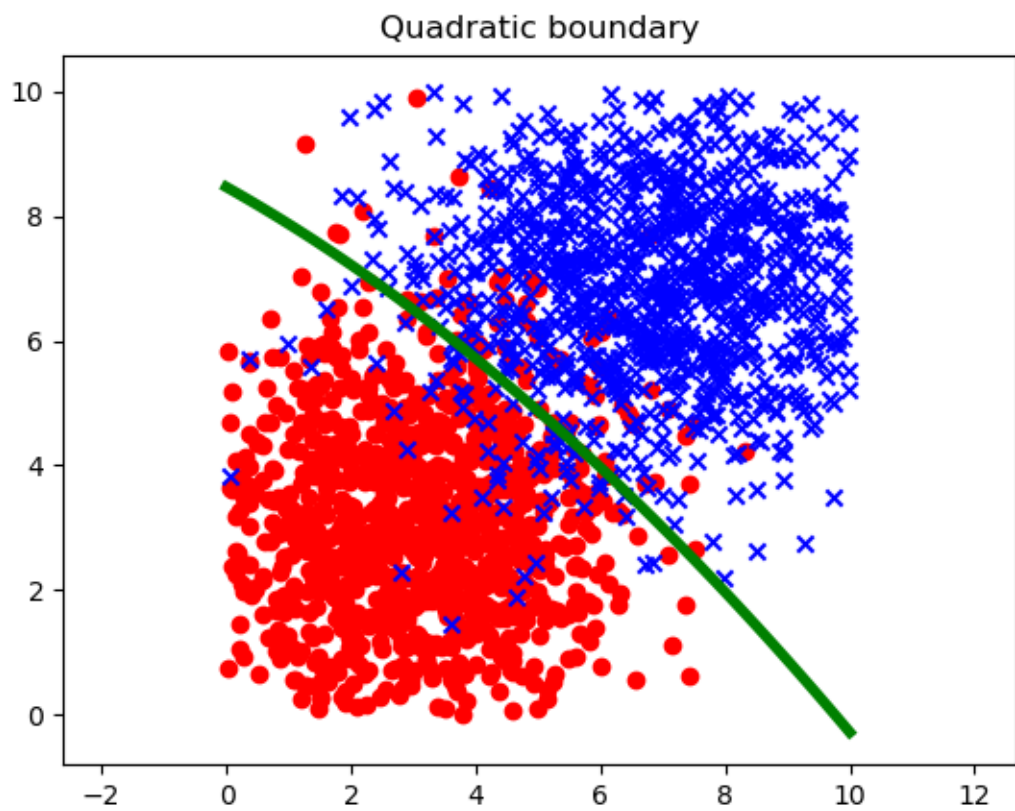
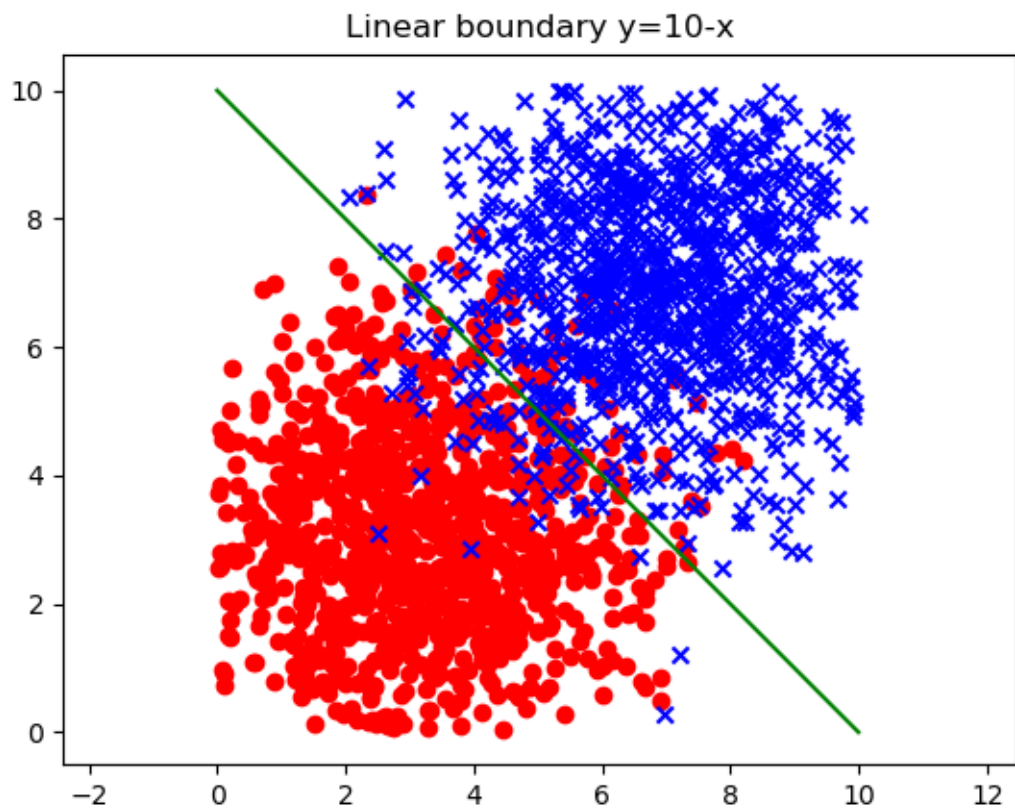
$$\Rightarrow (x-3)^2 - \frac{3}{4}(x-7)^2 = (y-7)^2 - (y-3)^2$$

$$= (y-7-y+3)(2y-10)$$

$$\frac{4(x-3)^2 - 3(x-7)^2}{4} = -8(y-5)$$

$$\Rightarrow \boxed{y = 5 - \frac{1}{32} (4(x-3)^2 - 3(x-7)^2)}$$

The generated plots are:



Q2a)

Ans-2 a) The dimensions of  $D$  are  $2000 \times 200$ .  
As there are about 35 different types of students  
so practical rank is 35.  
Now if  $D_{ij}$  is the number of times student  $i$  orders  
from restaurant  $j$  then we can have practical  
rank of  $\text{dmax } 200$ . Hence by this rank increases.

Q2b)

b) Now from data we can map the  
info of new students like which student  
orders from which restaurants and how many times.  
Now we will classify these new students as  
one of the 35 categories using  $k$  nearest neighbours  
algorithm.

The KNN algorithm is described as:

Calculate distance of sample point from all training points  
sort the distances and select the nearest  $k$  points  
Depending on the majority of a class of points in neighbourhood  
classify the sample point in one of the classes.

Q2c)



c) Now we have to recommend students to a restaurant. Let us define a restaurant as  $d \times 1$  vector with  $d$  type of menu items. With SVD we will find rank of this and find the categories of restaurants. Now based on restaurant category & transaction details we can recommend students to a restaurant by using the KNN algorithm. By finding the category of restaurant & thus recommending students which order from similar restaurants.

The KNN algorithm is described as:

- Calculate distance of sample point from all training points
- Sort the distances and select the nearest  $k$  points
- Depending on the majority of a class of points in neighborhood classify the sample point in one of the classes.

Q3)

Pr-3

By definition a positive semi-definite matrix is such  
 $u^T \Sigma u \geq 0 \quad \forall \text{ } n \times 1 \text{ vector } u.$

$$\Sigma = \text{Expected} ((X-y)(X-y)^T) = E((X-y)(X-y)^T)$$

$$u^T \Sigma u = E(u^T(X-y)(u^T(X-y))^T) = E(y^2)$$

$$\text{where } y = u^T(X-y)$$

Now expected value of positive quantity is positive  
so  $E(y^2) \geq 0 \Rightarrow u^T \Sigma u \geq 0$

so  $\Sigma$  is positive semi-definite.