Problem 1

Write a function (in Python or any language of your choice) to calculate the discriminant function for the given normal density equation (as given below) and prior probabilities.

Solution :-

We are provided with the formula for discriminant function as –

$$g_i(\mathrm{x}) = -\frac{1}{2}(\mathrm{x}-\mu_i)^t\Sigma_i^{-1}(\mathrm{x}-\mu_i) - \frac{d}{2}\ln 2\pi - \frac{1}{2}\ln|\Sigma_i| + \ln P(\omega_i).$$

Code for the given formula is given by :-

```python
from math import pi, log
import numpy as np

#applying the gi(x) formula on given dataset
#Discriminant Function
def dis_func(x, prob, mean, cov, d):
    temp1 = np.matmul( (x - mean).T,np.matmul(0.5*(x - mean), np.linalg.inv(cov)))
    temp2=0.5*d*log(2*pi)
    temp3=0.5*log(np.linalg.det(cov))
    temp4=(log(prob) if prob != 0 else 0)

    ans=-temp1-temp2-temp3+temp4
    return ans
```

Here the inputs for the function are :-

x - d-component column vector
prob - prior probability
mean- d-component mean vector
cov - d-component covariance vector
d - dimension of input sample

Input :-

```
input_data = [
    np.array([
        [-5.01, -8.12, -3.68],
        [-5.43, -3.48, -3.54],
        [1.08, -5.52, 1.66],
        [0.86, -3.78, -4.11],
        [-2.67, 0.63, 7.39],
        [4.94, 3.29, 2.08],
        [-2.51, 2.09, -2.59],
        [-2.25, -2.13, -6.94],
        [5.56, 2.86, -2.26],
        [1.03, -3.33, 4.33]
    ]),
    np.array([
        [-0.91, -0.18, -0.05],
        [1.30, -2.06, -3.53],
        [-7.75, -4.54, -0.95],
        [-5.47, 0.50, 3.92],
        [6.14, 5.72, -4.85],
        [3.60, 1.26, 4.36],
        [5.37, -4.63, -3.65],
        [7.18, 1.46, -6.66],
        [-7.39, 1.17, 6.30],
        [-7.50, -6.32, -0.31]

    ]),
    np.array([
        [5.35, 2.26, 8.13],
        [5.12, 3.22, -2.66],
        [-1.34, -5.31, -9.87],
        [4.48, 3.42, 5.19],
        [7.11, 2.39, 9.21],
        [7.17, 4.33, -0.98],
        [5.75, 3.97, 6.65],
        [0.77, 0.27, 2.41],
        [0.90, -0.43, -8.71],
        [3.52, -0.36, 6.43]
    ])
]
```

This inputs are taken from question and stored in input data array. We run a loop through all samples. For all we also find the class which gives maximum output from our discriminant function.

We assumed equal prior probability for each class of dataset.

Also to maintain rates of failure and success from the function we are taking count and total_count. We have found success rate for each case.

Output :-

```
Data classes should be classified as: 1
[-5.01 -8.12 -3.68]             classified as 1
[-5.43 -3.48 -3.54]             classified as 1
[ 1.08 -5.52  1.66]             classified as 1
[ 0.86 -3.78 -4.11]             classified as 1
[-2.67  0.63  7.39]             classified as 2
[4.94 3.29 2.08]                classified as 3
[-2.51  2.09 -2.59]             classified as 1
[-2.25 -2.13 -6.94]             classified as 1
[ 5.56  2.86 -2.26]             classified as 3
[ 1.03 -3.33  4.33]             classified as 1
Rate of Success: 70.0 %
Rate of Failure: 30.0 %

Data classes should be classified as: 2
[-0.91 -0.18 -0.05]             classified as 2
[ 1.3  -2.06 -3.53]             classified as 3
[-7.75 -4.54 -0.95]             classified as 2
[-5.47  0.5   3.92]             classified as 2
[ 6.14  5.72 -4.85]             classified as 2
[3.6  1.26 4.36]                classified as 3
[ 5.37 -4.63 -3.65]             classified as 2
[ 7.18  1.46 -6.66]             classified as 2
[-7.39  1.17  6.3 ]             classified as 2
[-7.5  -6.32 -0.31]             classified as 2
Rate of Success: 80.0 %
Rate of Failure: 20.0 %

Data classes should be classified as: 3
[5.35 2.26 8.13]                classified as 3
[ 5.12  3.22 -2.66]             classified as 3
[-1.34 -5.31 -9.87]             classified as 3
[4.48 3.42 5.19]                classified as 3
[7.11 2.39 9.21]                classified as 3
[ 7.17  4.33 -0.98]             classified as 3
[5.75 3.97 6.65]                classified as 3
[0.77 0.27 2.41]                classified as 1
[ 0.9  -0.43 -8.71]             classified as 3
[ 3.52 -0.36  6.43]             classified as 3
Rate of Success: 90.0 %
Rate of Failure: 10.0 %


...Program finished with exit code 0
Press ENTER to exit console.
```