ML ASSIGNMENT

Name - Ritik Gautam
Roll No - B180630CS

Discriminant Function

Q. Write a function to calculate the discriminant function for the given normal density equation (as given below) and prior probabilities.

$$g_i(x) = -\frac{1}{2}(x-\mu_i)^t \sum_i^{-1}(x-\mu_i) - \frac{d}{2} ln\, 2\pi - \frac{1}{2} ln \mid \sum_i \mid + ln\, P(\omega_i)$$

Solution:
We are given the sample data. Each column defines sample($\omega 1$,$\omega 2$,$\omega 3$) and every column consists of 10 samples. For every column of samples there are three features($x1$, $x2$ and $x3$). To classify the data points we will use above discriminant function.

```
# Import Libraries that we need
import numpy as np
from numpy import log
from numpy.linalg import inv as inverse, det as determinant
```

```
# Given input
n = 3
data = [
        # ω1
        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]
        ]),
        # ω2
        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]

        ]),

        # ω3
        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]
        ])
    ]
```

```
#find mean
mean = []
for i in range(len(data)):
  mean.append([sum(x)/len(x) for x in zip(*data[i])])
mean = np.array(mean)

#find covariance
covariance = []
for i in range(len(data)):
    covariance.append(np.cov(data[i].T))
covariance = np.array(covariance)

P = [1/2, 1/2, 0]


#The discriminant function is given below
def discriminant_function(i: int, x: np.array, P: list):
    if P[i] == 0:
        return -np.inf

    # finding dimension of input x
    dimention = x.shape[0]
    # Get the mean values based on given dimention
    mean_dimention = mean[:, 0:dimention]
    # Get the covariance values based on given dimention
    covariance_dimention = covariance[:, 0:dimention, 0:dimention]
    temp = np.matmul(inverse(covariance_dimention[i]), (x - mean_dimention[i]))
    #discriminant function
    res = -0.5 * np.matmul((x - mean_dimention[i]).T, temp) -0.5 * dimention * log(2 * np.pi)\
     - 0.5 * log(determinant(covariance_dimention[i]))
     + log(P[i])
    return res


# Now, check for all data points using above discriminant function
for i in range(n):
    print("\n")
    print("CLASS ω%d" % (i + 1))
    print("."*12)
```

```
for x in data[i]:
    print("Discriminant values for", x, end='\t')
    for i in range(n):
        print("ω%d : %.3f\t" % (i+1, discriminant_function(i, x, P)), end=' ')
    print()
```

```
CLASS ω1
............
Discriminant values for [-5.01 -8.12 -3.68]     ω1 : -8.897      ω2 : -9.888     ω3 : -inf
Discriminant values for [-5.43 -3.48 -3.54]     ω1 : -8.380      ω2 : -9.087     ω3 : -inf
Discriminant values for [ 1.08 -5.52  1.66]     ω1 : -8.696      ω2 : -10.442    ω3 : -inf
Discriminant values for [ 0.86 -3.78 -4.11]     ω1 : -8.118      ω2 : -8.204     ω3 : -inf
Discriminant values for [-2.67  0.63  7.39]     ω1 : -10.040     ω2 : -9.839     ω3 : -inf
Discriminant values for [4.94 3.29 2.08]        ω1 : -8.665      ω2 : -9.084     ω3 : -inf
Discriminant values for [-2.51  2.09 -2.59]     ω1 : -8.838      ω2 : -9.252     ω3 : -inf
Discriminant values for [-2.25 -2.13 -6.94]     ω1 : -8.447      ω2 : -10.148    ω3 : -inf
Discriminant values for [ 5.56  2.86 -2.26]     ω1 : -9.080      ω2 : -8.212     ω3 : -inf
Discriminant values for [ 1.03 -3.33  4.33]     ω1 : -8.454      ω2 : -10.700    ω3 : -inf


CLASS ω2
............
Discriminant values for [-0.91 -0.18 -0.05]     ω1 : -7.585      ω2 : -7.544     ω3 : -inf
Discriminant values for [ 1.3  -2.06 -3.53]     ω1 : -7.847      ω2 : -7.821     ω3 : -inf
Discriminant values for [-7.75 -4.54 -0.95]     ω1 : -9.411      ω2 : -8.814     ω3 : -inf
Discriminant values for [-5.47  0.5   3.92]     ω1 : -10.230     ω2 : -8.237     ω3 : -inf
Discriminant values for [ 6.14  5.72 -4.85]     ω1 : -10.854     ω2 : -9.857     ω3 : -inf
Discriminant values for [3.6  1.26 4.36]        ω1 : -8.415      ω2 : -10.040    ω3 : -inf
Discriminant values for [ 5.37 -4.63 -3.65]     ω1 : -10.644     ω2 : -9.662     ω3 : -inf
Discriminant values for [ 7.18  1.46 -6.66]     ω1 : -11.191     ω2 : -8.754     ω3 : -inf
Discriminant values for [-7.39  1.17  6.3 ]     ω1 : -12.933     ω2 : -9.077     ω3 : -inf
Discriminant values for [-7.5  -6.32 -0.31]     ω1 : -9.361      ω2 : -8.897     ω3 : -inf


CLASS ω3
............
Discriminant values for [5.35 2.26 8.13]        ω1 : -9.992      ω2 : -14.435    ω3 : -inf
Discriminant values for [ 5.12  3.22 -2.66]     ω1 : -9.073      ω2 : -8.272     ω3 : -inf
Discriminant values for [-1.34 -5.31 -9.87]     ω1 : -9.798      ω2 : -11.734    ω3 : -inf
Discriminant values for [4.48 3.42 5.19]        ω1 : -9.027      ω2 : -10.732    ω3 : -inf
Discriminant values for [7.11 2.39 9.21]        ω1 : -11.057     ω2 : -17.362    ω3 : -inf
```

```
Discriminant values for [ 7.17  4.33 -0.98]     ω1 : -9.819      ω2 : -8.902     ω3 : -inf
Discriminant values for [5.75 3.97 6.65]        ω1 : -9.774      ω2 : -12.636    ω3 : -inf
Discriminant values for [0.77 0.27 2.41]        ω1 : -7.738      ω2 : -8.149     ω3 : -inf
Discriminant values for [ 0.9  -0.43 -8.71]     ω1 : -9.434      ω2 : -10.707    ω3 : -inf
Discriminant values for [ 3.52 -0.36  6.43]     ω1 : -9.027      ω2 : -12.419    ω3 : -inf
```

✓ 0s completed at 12:44 PM