Homework 01 - Multivariate Parametric Classification

Generating Random Points

- After importing NumPy, Pandas and Matplotlib, I started by generating random points with the provided parameters. I used NumPy's Random module's **multivariate_normal** function, and filled its *mean*, *cov*, and *size* parameters. Then using NumPy's vstack and concatenate functions, I have created X (data matrix) and y (outputs).
- Then I plotted the data to make sure that it looks similar to the one on the guideline.

Estimating the Parameters

- To estimate sample means, sample covariance matrices and prior probabilities, I used the following formulas from the 5th chapter of the book:

Sample Mean

Sample mean, $\mathbf{m}: m_i = \frac{\sum_{t=1}^{N} x_i^t}{N}, i = 1, \ldots, d$

Sample Covariance Matrix

Sample covariance matrix, $\mathbf{S}: s_{ij} = \frac{\sum_{t=1}^{N}(x_i^t - m_i)(x_j^t - m_j)}{N}$

Prior Probabilities

$\hat{P}(C_i) = \frac{\sum_t r_i^t}{N}$

- I've written "estimate_sample_mean", "estimate_covariance_matrix", and "prior_probability" functions in order to apply each of them separately on the data I have generated previously.

Calculating Confusion Matrix

- Then, I had to develop a classification algorithm in order to make predictions and calculate the confusion matrix. To develop a classification algorithm, I used the **quadratic discriminant** formula from the book (which we saw in the lecture as **score function**):

Quadratic Discriminant

Score function: $g_i(\mathbf{x}) = -\frac{1}{2} log|\mathbf{S}_i| - \frac{1}{2}(\mathbf{x} - \mathbf{m}_i)^T \mathbf{S}_i^{-1}(\mathbf{x} - \mathbf{m}_i) + log\hat{P}(C_i)$

- I have created a "quadratic_discriminant" function in which I could place the sample mean, sample covariance matrix and prior probability I have calculated earlier.
- Lastly, I have written a "predict" function, which calculates scores for each class, assigns them to variables "g1", "g2", and "g3", and depending on which is greater, returns either class 1, 2, or 3. I applied the "predict" function for all the data points in my data matrix, and appended the predictions into a variable called "y_pred".
- By using Pandas's crosstab function, I have created a confusion matrix. According to my confusion matrix, I was able to correctly classify 254 data points, while 6 of them were misclassified.

Drawing Decision Boundaries

- Lastly, I in order to draw the decision boundaries, I have used another representation of the quadratic discriminant function from the book (and also from the lectures):

Quadratic Discriminant

$g_i(x) = x^T W_i x + w_i^T x + w_{i0}$

where

$W_i = -\frac{1}{2} S_i^{-1}$

$w_i = S_i^{-1} m_i$

$w_{i0} = -\frac{1}{2} m_i^T S_i^{-1} m_i - \frac{1}{2} log|S_i| + log\hat{P}(C_i)$

- Although I was able to calculate W_i, w_i and w_{i0}, and put a circle around the misclassified data points, I was unsuccessful at all my attempts at drawing the decision boundaries.