ENGR 421/DASC 521: Introduction to Machine Learning
Homework 2: Multivariate Parametric Classification
Deadline: April 5, 2023, 11:59 PM

In this homework, you will implement a multivariate parametric classification using Python. Here are the steps you need to follow:

1. Read Chapter 5 from the textbook.

2. You are given a multivariate classification data set, which contains 5000 data points from a two-dimensional feature space. These data points are from five distinct classes, where we have 1000 data points from each class. You are provided with two data files:

   a. `hw02_data_points.csv`: two-dimensional data points,

   b. `hw02_class_labels.csv`: corresponding class labels.

3. Calculate the prior probability estimates $\widehat{\Pr}(y = 1), \widehat{\Pr}(y = 2), \dots, \widehat{\Pr}(y = 5)$ using the training data points. (10 points)

```
class_priors = estimate_prior_probabilities(y_train)
print(class_priors)

[0.2 0.2 0.2 0.2 0.2]
```

Hint: You can use the following equation to calculate the prior probability estimates.

$$\widehat{\Pr}(y = c) = \frac{\sum_{i=1}^{N} 1(y_i = c)}{N} = \frac{N_c}{N}$$

---

4. Calculate the class mean estimates $\hat{\boldsymbol{\mu}}_1, \hat{\boldsymbol{\mu}}_2, \dots, \hat{\boldsymbol{\mu}}_5$ using the training data points. (20 points)

```
sample_means = estimate_class_means(X_train, y_train)
print(sample_means)

[[ -6.64451313 -26.36348034]
 [-42.59684357  -3.08704541]
 [-15.33132145  34.74988518]
 [ 35.28039812  28.29476758]
 [ 29.29228003 -33.59412701]]
```

Hint: You can use the following equation to calculate the class mean estimates.

$$\hat{\boldsymbol{\mu}}_c = \frac{\sum_{i=1}^{N} \boldsymbol{x}_i 1(y_i = c)}{\sum_{i=1}^{N} 1(y_i = c)}$$

---

5. Calculate the class covariance estimates $\hat{\mathbf{\Sigma}}_1, \hat{\mathbf{\Sigma}}_2, \ldots, \hat{\mathbf{\Sigma}}_5$ using the training data points. (20 points)

```
sample_covariances = estimate_class_covariances(X_train, y_train)
print(sample_covariances)
```

```
[[[ 268.24169454   84.38622865]
  [  84.38622865  165.60007039]]
 [[ 268.36399098  -79.36361871]
  [ -79.36361871  228.81216241]]
 [[ 257.88530822  107.48459802]
  [ 107.48459802  270.90303479]]
 [[ 390.64688372 -143.01194574]
  [-143.01194574  159.85719588]]
 [[  62.29030005    8.10502983]
  [   8.10502983  379.25858684]]]
```

Hint: You can use the following equation to calculate the class covariance estimates.

$$\hat{\mathbf{\Sigma}}_c = \frac{\sum_{i=1}^{N} (\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_c)(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_c)^\top 1(y_i = c)}{\sum_{i=1}^{N} 1(y_i = c)}$$

---

6. Calculate the score values for the data points in your training and test sets using the estimated parameters. (30 points)

```
scores_train = calculate_score_values(X_train, sample_means,
                                      sample_covariances, class_priors)
print(scores_train)
```

```
[[-14.19538107 -22.10065254 -32.17093002 -22.95654712  -9.69739781]
 [-13.31824343 -21.10515229 -30.34378865 -22.68609467  -9.16710182]
 [-15.84197823 -20.97522186 -36.85943093 -35.44103047  -8.93068396]
 ...
 [-21.2864439  -33.52980121 -17.3443919   -9.47618622 -16.51638154]
 [-15.17110159 -24.16805014 -16.72250881  -9.40819221 -11.97010383]
 [-20.31293361 -31.85967687 -16.98927511  -9.1391833  -15.61253304]]
```

Hint: You can use the following equation to calculate the score values.

$$\begin{aligned} g_c(\boldsymbol{x}) &= \log \hat{p}(\boldsymbol{x}|y = c) + \log \widehat{\Pr}(y = c) \\ &= -\frac{D}{2}\log(2\pi) - \frac{1}{2}\log(|\hat{\mathbf{\Sigma}}_c|) - \frac{1}{2}(\boldsymbol{x} - \hat{\boldsymbol{\mu}}_c)^\top \hat{\mathbf{\Sigma}}_c^{-1}(\boldsymbol{x} - \hat{\boldsymbol{\mu}}_c) + \log \widehat{\Pr}(y = c) \end{aligned}$$

---

7. Calculate the confusion matrix for the training data points using the calculated score values. (10 points)

```
confusion_train = calculate_confusion_matrix(y_train, scores_train)
print(confusion_train)
```

```
[[829 136   0   0  37]
 [ 46 785 147   0   0]
 [  0  79 791 135   0]
 [  0   0  62 865   0]
 [125   0   0   0 963]]
```

8. Calculate the shared covariance estimate $\hat{\mathbf{\Sigma}}_1 = \hat{\mathbf{\Sigma}}_2 = \cdots = \hat{\mathbf{\Sigma}}_5 = \hat{\mathbf{\Sigma}}$ using the training data points. (10 points)

```
sample_covariances = estimate_shared_class_covariance(X_train, y_train)
print(sample_covariances)
```

```
[[[1088.7724787   -46.85767937]
  [ -46.85767937 1009.14155144]]
 [[1088.7724787   -46.85767937]
  [ -46.85767937 1009.14155144]]
 [[1088.7724787   -46.85767937]
  [ -46.85767937 1009.14155144]]
 [[1088.7724787   -46.85767937]
  [ -46.85767937 1009.14155144]]
 [[1088.7724787   -46.85767937]
  [ -46.85767937 1009.14155144]]]
```

```
scores_train = calculate_score_values(X_train, sample_means,
                                      sample_covariances, class_priors)
print(scores_train)
```

```
[[-11.46793996 -13.90222671 -13.76045065 -12.04097548 -10.48248827]
 [-11.32991837 -13.62530355 -13.50637918 -11.94332728 -10.45947009]
 [-11.34232183 -13.92476829 -14.67342885 -13.22089383 -10.49857077]
 ...
 [-13.6968323  -14.84896905 -12.0931904  -10.47805967 -12.95656273]
 [-12.1627728  -13.33771435 -11.61361575 -10.46378924 -11.7028491 ]
 [-13.43558046 -14.56759373 -11.96475973 -10.44324505 -12.75521448]]
```

```
confusion_train = calculate_confusion_matrix(y_train, scores_train)
print(confusion_train)
```

```
[[833 142   0   0   3]
 [ 26 836 174   0   0]
 [  0  22 804 148   0]
 [ 38   0  22 852  69]
 [103   0   0   0 928]]
```

Hint: You can use the following equations to calculate the shared covariance estimate.

$$\hat{\boldsymbol{\mu}} = \frac{\sum\limits_{i=1}^{N} \boldsymbol{x}_i}{N}$$

$$\hat{\boldsymbol{\Sigma}}_1 = \hat{\boldsymbol{\Sigma}}_2 = \cdots = \hat{\boldsymbol{\Sigma}}_5 = \hat{\boldsymbol{\Sigma}} = \frac{\sum\limits_{i=1}^{N} (\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})^\top}{N}$$

What to submit: You need to submit your source code in a single file (`.py` file). You are provided with a template file named as `0099999.py`, where `99999` should be replaced with your 5-digit student number. You are allowed to change the template file between the following lines.

```
# your implementation starts below

# your implementation ends above
```

How to submit: Submit the file you edited to Blackboard by following the exact style mentioned. Submissions that do not follow these guidelines will not be graded.

Late submission policy: Late submissions will not be graded.

Cheating policy: Very similar submissions will not be graded.
