Machine Learning

Homework 1

系級：醫工三　　學號：B12508025　　姓名：盧雅筠

Problem 1: cross-validation, feature selection, and classification

I. Reports

1. The multivariate Gaussian Distribution models and Parameter Estimation

- For each class $y \in \{0,1\}$, $\mathbf{x} \in \mathbb{R}^d$ assume the multivariate Gaussian distribution:

$$\mathbf{x} \mid y = c \sim \mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c), \quad c \in \{0,1\}$$

$$p(\mathbf{x} \mid y = c) = \frac{1}{(2\pi)^{d/2} \mid \boldsymbol{\Sigma}_c \mid^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_c)^T \boldsymbol{\Sigma}_c^{-1}(\mathbf{x} - \boldsymbol{\mu}_c)\right]$$

where $\boldsymbol{\mu}_c$ and $\boldsymbol{\Sigma}_c$ denote the mean vector and covariance matrix of class c.

- For 單一樣本 x, log-likelihood 如下:

$$\log p(x \mid \mu, \Sigma) = -\frac{1}{2}\left[d \log(2\pi) + \log\det(\Sigma) + (x - \mu)^\top \Sigma^{-1}(x - \mu)\right]$$

Cholesky decomposition of the covariance matrix: $\Sigma = LL^\top$，這個表示法可以使數值穩定，跳過反矩陣運算，得 $\log\det(\Sigma) = 2\sum_{i=1}^{d} \log L_{ii}, \quad (x - \mu)^\top \Sigma^{-1}(x - \mu) = \|L^{-1}(x - \mu)\|_2^2$

- Parameter Estimation: maximum likelihood estimation (MLE) use

`np.mean(X_c, axis=0)` and `cov = np.cov(X_c, rowvar=False)` and `MLE_Estimater()` to compute:

$$\ell(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) = \sum_{i:y_i=c} \log p(\mathbf{x}_i \mid \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$$

$$\hat{\boldsymbol{\mu}}_c = \frac{1}{N_c} \sum_{i:y_i=c} \mathbf{x}_i \qquad \hat{\boldsymbol{\Sigma}}_c = \frac{1}{N_c - 1} \sum_{i:y_i=c} (\mathbf{x}_i - \hat{\boldsymbol{\mu}}_c)(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_c)^T$$

- Posterior Probability: use

`Multivariate_Gaussian_Distribution_log_likelihood()` and `Bayesian_decision_classifier()` to compute:

$$\ell_0 = \log p(x \mid y = 0) + \log p_0$$
$$\ell_1 = \log p(x \mid y = 1) + \log p_1$$

$$P(y = 1 \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid y = 1)p(y = 1)}{p(\mathbf{x} \mid y = 1)p(y = 1) + p(\mathbf{x} \mid y = 0)p(y = 0)} = \frac{e^{\ell_1}}{e^{\ell_0} + e^{\ell_1}} = \frac{1}{1 + e^{-(\ell_1 - \ell_0)}}$$

- Discriminant Function: use

  `Multivariate_Gaussian_Distribution_log_likelihood()` and **`Bayesian_decision_classifier()`** to compute:

$$g_c(\mathbf{x}) = \ln P(\mathbf{x} \mid y = c) + \ln p(y = c)$$

$\Delta(\mathbf{x}) = g_1(\mathbf{x}) - g_0(\mathbf{x})$, the decision boundary is $\Delta(\mathbf{x}) = 0$

$$\Delta \overset{\text{def}}{=} \ell_1 - \ell_0 = \log \frac{P(y = 1 \mid x)}{P(y = 0 \mid x)}$$

  為了避免 overflow，使用 log-sum-exp trick：$m = \max(\ell_0, \ell_1)$

$$P(y = 1 \mid x) = \frac{e^{\ell_1 - m}}{e^{\ell_0 - m} + e^{\ell_1 - m}}$$

2. Forward Feature Selection and its cost Function

- Feature selection:

  use **`Forward_Selection_AUC()`** and its cost function **`cv_AUC_score()`**

  Starting from an empty feature set, at each iteration the algorithm tests all remaining features and adds the one that maximizes the AUC score. **`cv_AUC_score()`** use Stratified K-Fold calculate AUC score.

- Determine the number of features to be selected

  The maximum features are determined by

$$\max\_features = \min\left(\max(1, \min(n_0, n_1) - 1), d\right)$$

  where n_0, n_1 is the number of two classes samples, d is X_{train} . shape[1]

3. Selected Features per Fold

- Each of the 103 data samples serves as a test point once in leave-one-out cross-validation, yielding 103 different feature subsets. The selected features for each fold are stored in **`selected_features`**. The list of selected features per fold is shown in Figure 1

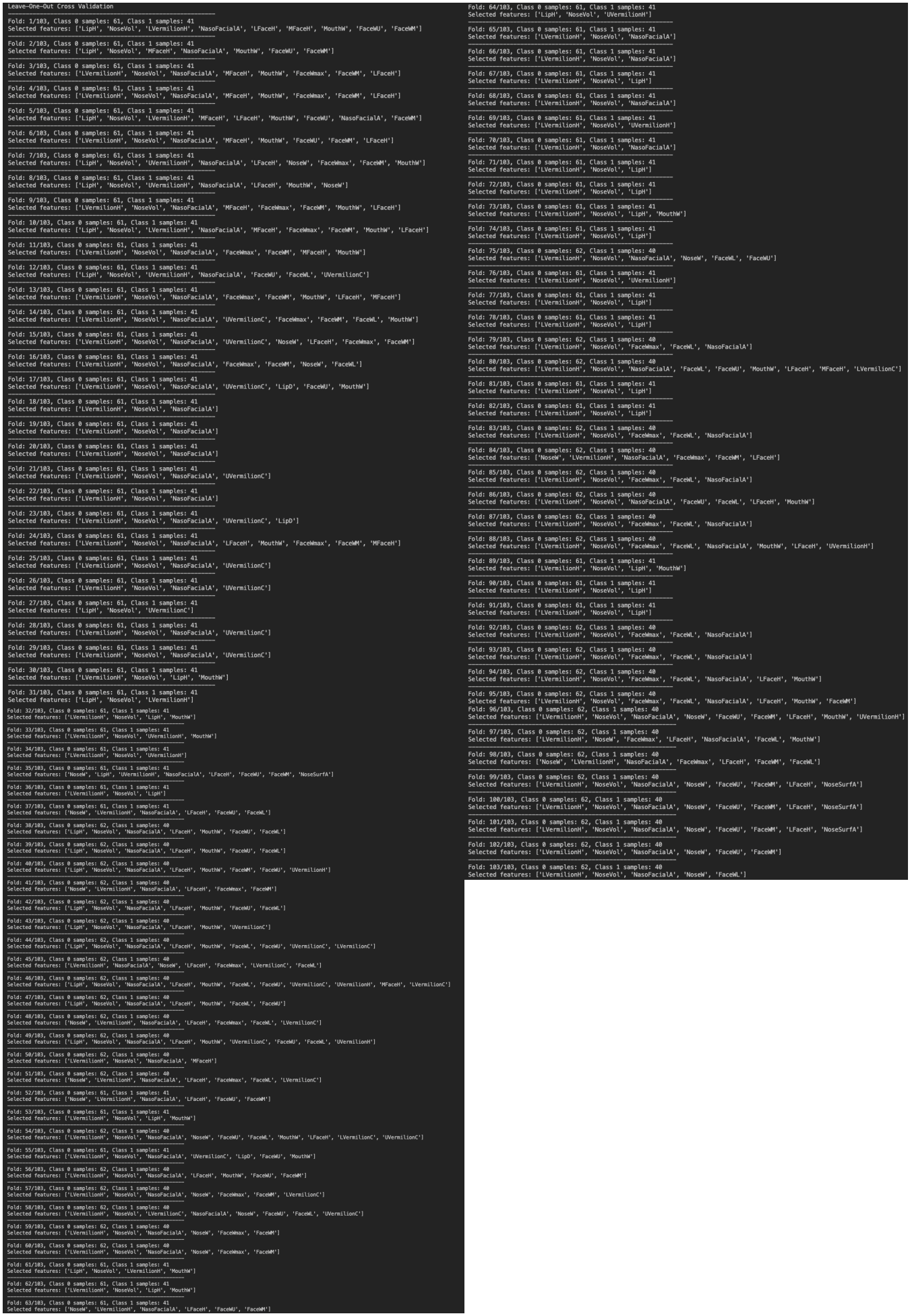

Leave-One-Out Cross Validation
Fold: 1/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LipH', 'NoseVol', 'LVermilionH', 'NasoFacialA', 'LFaceH', 'MFaceH', 'MouthW', 'FaceWU', 'FaceWM']
Fold: 2/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LipH', 'NoseVol', 'MFaceH', 'NasoFacialA', 'MouthW', 'FaceWU', 'FaceWM']
Fold: 3/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LVermilionH', 'NoseVol', 'NasoFacialA', 'MFaceH', 'MouthW', 'FaceWmax', 'FaceWM', 'LFaceH']
Fold: 4/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LVermilionH', 'NoseVol', 'NasoFacialA', 'MFaceH', 'MouthW', 'FaceWmax', 'FaceWM', 'LFaceH']
Fold: 5/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LipH', 'NoseVol', 'LVermilionH', 'MFaceH', 'LFaceH', 'MouthW', 'FaceWU', 'NasoFacialA', 'FaceWM']
Fold: 6/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LVermilionH', 'NoseVol', 'NasoFacialA', 'MFaceH', 'MouthW', 'FaceWU', 'FaceWM', 'LFaceH']
Fold: 7/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LipH', 'NoseVol', 'UVermilionH', 'NasoFacialA', 'LFaceH', 'NoseW', 'FaceWmax', 'FaceWM', 'MouthW']
Fold: 8/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LipH', 'NoseVol', 'UVermilionH', 'NasoFacialA', 'LFaceH', 'MouthW', 'NoseW']
Fold: 9/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LVermilionH', 'NoseVol', 'NasoFacialA', 'MFaceH', 'FaceWmax', 'FaceWM', 'MouthW', 'LFaceH']
Fold: 10/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LipH', 'NoseVol', 'LVermilionH', 'NasoFacialA', 'MFaceH', 'FaceWmax', 'FaceWM', 'MouthW', 'LFaceH']
Fold: 11/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LVermilionH', 'NoseVol', 'NasoFacialA', 'FaceWmax', 'FaceWM', 'MFaceH', 'MouthW']
Fold: 12/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LipH', 'NoseVol', 'UVermilionH', 'NasoFacialA', 'FaceWU', 'FaceWL', 'UVermilionC']
Fold: 13/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LVermilionH', 'NoseVol', 'NasoFacialA', 'FaceWmax', 'FaceWM', 'MouthW', 'LFaceH', 'MFaceH']
Fold: 14/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LVermilionH', 'NoseVol', 'NasoFacialA', 'UVermilionC', 'FaceWmax', 'FaceWM', 'FaceWL', 'MouthW']
Fold: 15/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LVermilionH', 'NoseVol', 'NasoFacialA', 'UVermilionC', 'NoseW', 'LFaceH', 'FaceWmax', 'FaceWM']
Fold: 16/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LVermilionH', 'NoseVol', 'NasoFacialA', 'FaceWmax', 'FaceWM', 'NoseW', 'FaceWL']
Fold: 17/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LVermilionH', 'NoseVol', 'NasoFacialA', 'UVermilionC', 'LipD', 'FaceWU', 'MouthW']
Fold: 18/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LVermilionH', 'NoseVol', 'NasoFacialA']
Fold: 19/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LVermilionH', 'NoseVol', 'NasoFacialA']
Fold: 20/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LVermilionH', 'NoseVol', 'NasoFacialA']
Fold: 21/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LVermilionH', 'NoseVol', 'NasoFacialA', 'UVermilionC']
Fold: 22/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LVermilionH', 'NoseVol', 'NasoFacialA']
Fold: 23/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LVermilionH', 'NoseVol', 'NasoFacialA', 'UVermilionC', 'LipD']
Fold: 24/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LVermilionH', 'NoseVol', 'NasoFacialA', 'LFaceH', 'MouthW', 'FaceWmax', 'FaceWM', 'MFaceH']
Fold: 25/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LVermilionH', 'NoseVol', 'NasoFacialA', 'UVermilionC']
Fold: 26/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LVermilionH', 'NoseVol', 'NasoFacialA', 'UVermilionC']
Fold: 27/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LipH', 'NoseVol', 'UVermilionC']
Fold: 28/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LVermilionH', 'NoseVol', 'NasoFacialA', 'UVermilionC']
Fold: 29/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LVermilionH', 'NoseVol', 'NasoFacialA', 'UVermilionC']
Fold: 30/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LVermilionH', 'NoseVol', 'LipH', 'MouthW']
Fold: 31/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LipH', 'NoseVol', 'LVermilionH']
Fold: 32/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LVermilionH', 'NoseVol', 'LipH', 'MouthW']
Fold: 33/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LVermilionH', 'NoseVol', 'UVermilionH', 'MouthW']
Fold: 34/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LVermilionH', 'NoseVol', 'UVermilionH']
Fold: 35/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['NoseW', 'LipH', 'UVermilionH', 'NasoFacialA', 'LFaceH', 'FaceWU', 'FaceWM', 'NoseSurfA']
Fold: 36/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LVermilionH', 'NoseVol', 'LipH']
Fold: 37/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['NoseW', 'LVermilionH', 'NasoFacialA', 'LFaceH', 'FaceWU', 'FaceWL']
Fold: 38/103, Class 0 samples: 62, Class 1 samples: 40
Selected features: ['LipH', 'NoseVol', 'NasoFacialA', 'LFaceH', 'MouthW', 'FaceWU', 'FaceWL']
Fold: 39/103, Class 0 samples: 62, Class 1 samples: 40
Selected features: ['LipH', 'NoseVol', 'NasoFacialA', 'LFaceH', 'MouthW', 'FaceWU', 'FaceWL']
Fold: 40/103, Class 0 samples: 62, Class 1 samples: 40
Selected features: ['LipH', 'NoseVol', 'NasoFacialA', 'LFaceH', 'MouthW', 'FaceWM', 'FaceWU', 'UVermilionH']
Fold: 41/103, Class 0 samples: 62, Class 1 samples: 40
Selected features: ['NoseW', 'LVermilionH', 'NasoFacialA', 'LFaceH', 'FaceWmax', 'FaceWM']
Fold: 42/103, Class 0 samples: 62, Class 1 samples: 40
Selected features: ['LipH', 'NoseVol', 'NasoFacialA', 'LFaceH', 'MouthW', 'FaceWU', 'FaceWL']
Fold: 43/103, Class 0 samples: 62, Class 1 samples: 40
Selected features: ['LipH', 'NoseVol', 'NasoFacialA', 'LFaceH', 'MouthW', 'UVermilionC']
Fold: 44/103, Class 0 samples: 62, Class 1 samples: 40
Selected features: ['LipH', 'NoseVol', 'NasoFacialA', 'LFaceH', 'MouthW', 'FaceWL', 'FaceWU', 'UVermilionC', 'LVermilionC']
Fold: 45/103, Class 0 samples: 62, Class 1 samples: 40
Selected features: ['LVermilionH', 'NasoFacialA', 'NoseW', 'LFaceH', 'FaceWmax', 'LVermilionC', 'FaceWL']
Fold: 46/103, Class 0 samples: 62, Class 1 samples: 40
Selected features: ['LipH', 'NoseVol', 'NasoFacialA', 'LFaceH', 'MouthW', 'FaceWL', 'FaceWU', 'UVermilionC', 'UVermilionH', 'MFaceH', 'LVermilionC']
Fold: 47/103, Class 0 samples: 62, Class 1 samples: 40
Selected features: ['LipH', 'NoseVol', 'NasoFacialA', 'LFaceH', 'MouthW', 'FaceWL', 'FaceWU']
Fold: 48/103, Class 0 samples: 62, Class 1 samples: 40
Selected features: ['NoseW', 'LVermilionH', 'NasoFacialA', 'LFaceH', 'FaceWmax', 'FaceWL', 'LVermilionC']
Fold: 49/103, Class 0 samples: 62, Class 1 samples: 40
Selected features: ['LipH', 'NoseVol', 'NasoFacialA', 'LFaceH', 'MouthW', 'UVermilionC', 'FaceWU', 'FaceWL', 'UVermilionH']
Fold: 50/103, Class 0 samples: 62, Class 1 samples: 40
Selected features: ['LVermilionH', 'NoseVol', 'NasoFacialA', 'MFaceH']
Fold: 51/103, Class 0 samples: 62, Class 1 samples: 40
Selected features: ['NoseW', 'LVermilionH', 'NasoFacialA', 'LFaceH', 'FaceWmax', 'FaceWL', 'LVermilionC']
Fold: 52/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['NoseW', 'LVermilionH', 'NasoFacialA', 'LFaceH', 'FaceWU', 'FaceWM']
Fold: 53/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LVermilionH', 'NoseVol', 'LipH', 'MouthW']
Fold: 54/103, Class 0 samples: 62, Class 1 samples: 40
Selected features: ['LVermilionH', 'NoseVol', 'NasoFacialA', 'NoseW', 'FaceWU', 'FaceWL', 'MouthW', 'LFaceH', 'LVermilionC', 'UVermilionC']
Fold: 55/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LVermilionH', 'NoseVol', 'NasoFacialA', 'UVermilionC', 'LipD', 'FaceWU', 'MouthW']
Fold: 56/103, Class 0 samples: 62, Class 1 samples: 40
Selected features: ['LVermilionH', 'NoseVol', 'NasoFacialA', 'LFaceH', 'MouthW', 'FaceWU', 'FaceWM']
Fold: 57/103, Class 0 samples: 62, Class 1 samples: 40
Selected features: ['LVermilionH', 'NoseVol', 'NasoFacialA', 'NoseW', 'FaceWmax', 'FaceWM', 'LVermilionC']
Fold: 58/103, Class 0 samples: 62, Class 1 samples: 40
Selected features: ['LVermilionH', 'NoseVol', 'LVermilionC', 'NasoFacialA', 'NoseW', 'FaceWU', 'FaceWL', 'UVermilionC']
Fold: 59/103, Class 0 samples: 62, Class 1 samples: 40
Selected features: ['LVermilionH', 'NoseVol', 'NasoFacialA', 'NoseW', 'FaceWmax', 'FaceWM']
Fold: 60/103, Class 0 samples: 62, Class 1 samples: 40
Selected features: ['LVermilionH', 'NoseVol', 'NasoFacialA', 'NoseW', 'FaceWmax', 'FaceWM']
Fold: 61/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LipH', 'NoseVol', 'LVermilionH', 'MouthW']
Fold: 62/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LVermilionH', 'NoseVol', 'LipH', 'MouthW']
Fold: 63/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['NoseW', 'LVermilionH', 'NasoFacialA', 'LFaceH', 'FaceWU', 'FaceWM']
Fold: 64/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LipH', 'NoseVol', 'UVermilionH']
Fold: 65/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LVermilionH', 'NoseVol', 'NasoFacialA']
Fold: 66/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LVermilionH', 'NoseVol', 'NasoFacialA']
Fold: 67/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LVermilionH', 'NoseVol', 'LipH']
Fold: 68/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LVermilionH', 'NoseVol', 'NasoFacialA']
Fold: 69/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LVermilionH', 'NoseVol', 'UVermilionH']
Fold: 70/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LVermilionH', 'NoseVol', 'NasoFacialA']
Fold: 71/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LVermilionH', 'NoseVol', 'LipH']
Fold: 72/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LVermilionH', 'NoseVol', 'LipH']
Fold: 73/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LVermilionH', 'NoseVol', 'LipH', 'MouthW']
Fold: 74/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LVermilionH', 'NoseVol', 'LipH']
Fold: 75/103, Class 0 samples: 62, Class 1 samples: 40
Selected features: ['LVermilionH', 'NoseVol', 'NasoFacialA', 'NoseW', 'FaceWL', 'FaceWU']
Fold: 76/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LVermilionH', 'NoseVol', 'UVermilionH']
Fold: 77/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LVermilionH', 'NoseVol', 'LipH']
Fold: 78/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LVermilionH', 'NoseVol', 'LipH']
Fold: 79/103, Class 0 samples: 62, Class 1 samples: 40
Selected features: ['LVermilionH', 'NoseVol', 'FaceWmax', 'FaceWL', 'NasoFacialA']
Fold: 80/103, Class 0 samples: 62, Class 1 samples: 40
Selected features: ['LVermilionH', 'NoseVol', 'NasoFacialA', 'FaceWL', 'FaceWU', 'MouthW', 'LFaceH', 'MFaceH', 'LVermilionC']
Fold: 81/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LVermilionH', 'NoseVol', 'LipH']
Fold: 82/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LVermilionH', 'NoseVol', 'LipH']
Fold: 83/103, Class 0 samples: 62, Class 1 samples: 40
Selected features: ['LVermilionH', 'NoseVol', 'FaceWmax', 'FaceWL', 'NasoFacialA']
Fold: 84/103, Class 0 samples: 62, Class 1 samples: 40
Selected features: ['NoseW', 'LVermilionH', 'NasoFacialA', 'FaceWmax', 'FaceWM', 'LFaceH']
Fold: 85/103, Class 0 samples: 62, Class 1 samples: 40
Selected features: ['LVermilionH', 'NoseVol', 'FaceWmax', 'FaceWL', 'NasoFacialA']
Fold: 86/103, Class 0 samples: 62, Class 1 samples: 40
Selected features: ['LVermilionH', 'NoseVol', 'NasoFacialA', 'FaceWU', 'FaceWL', 'LFaceH', 'MouthW']
Fold: 87/103, Class 0 samples: 62, Class 1 samples: 40
Selected features: ['LVermilionH', 'NoseVol', 'FaceWmax', 'FaceWL', 'NasoFacialA']
Fold: 88/103, Class 0 samples: 62, Class 1 samples: 40
Selected features: ['LVermilionH', 'NoseVol', 'FaceWmax', 'FaceWL', 'NasoFacialA', 'MouthW', 'LFaceH', 'UVermilionH']
Fold: 89/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LVermilionH', 'NoseVol', 'LipH', 'MouthW']
Fold: 90/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LVermilionH', 'NoseVol', 'LipH']
Fold: 91/103, Class 0 samples: 61, Class 1 samples: 41
Selected features: ['LVermilionH', 'NoseVol', 'LipH']
Fold: 92/103, Class 0 samples: 62, Class 1 samples: 40
Selected features: ['LVermilionH', 'NoseVol', 'FaceWmax', 'FaceWL', 'NasoFacialA']
Fold: 93/103, Class 0 samples: 62, Class 1 samples: 40
Selected features: ['LVermilionH', 'NoseVol', 'FaceWmax', 'FaceWL', 'NasoFacialA']
Fold: 94/103, Class 0 samples: 62, Class 1 samples: 40
Selected features: ['LVermilionH', 'NoseVol', 'FaceWmax', 'FaceWL', 'NasoFacialA', 'LFaceH', 'MouthW']
Fold: 95/103, Class 0 samples: 62, Class 1 samples: 40
Selected features: ['LVermilionH', 'NoseVol', 'FaceWmax', 'FaceWL', 'NasoFacialA', 'LFaceH', 'MouthW', 'FaceWM']
Fold: 96/103, Class 0 samples: 62, Class 1 samples: 40
Selected features: ['LVermilionH', 'NoseVol', 'NasoFacialA', 'NoseW', 'FaceWU', 'FaceWM', 'LFaceH', 'MouthW', 'UVermilionH']
Fold: 97/103, Class 0 samples: 62, Class 1 samples: 40
Selected features: ['LVermilionH', 'NoseW', 'FaceWmax', 'LFaceH', 'NasoFacialA', 'FaceWL', 'MouthW']
Fold: 98/103, Class 0 samples: 62, Class 1 samples: 40
Selected features: ['NoseW', 'LVermilionH', 'NasoFacialA', 'FaceWmax', 'LFaceH', 'FaceWM', 'FaceWL']
Fold: 99/103, Class 0 samples: 62, Class 1 samples: 40
Selected features: ['LVermilionH', 'NoseVol', 'NasoFacialA', 'NoseW', 'FaceWU', 'FaceWM', 'LFaceH', 'NoseSurfA']
Fold: 100/103, Class 0 samples: 62, Class 1 samples: 40
Selected features: ['LVermilionH', 'NoseVol', 'NasoFacialA', 'NoseW', 'FaceWU', 'FaceWM', 'LFaceH', 'NoseSurfA']
Fold: 101/103, Class 0 samples: 62, Class 1 samples: 40
Selected features: ['LVermilionH', 'NoseVol', 'NasoFacialA', 'NoseW', 'FaceWU', 'FaceWM', 'LFaceH', 'NoseSurfA']
Fold: 102/103, Class 0 samples: 62, Class 1 samples: 40
Selected features: ['LVermilionH', 'NoseVol', 'NasoFacialA', 'NoseW', 'FaceWU', 'FaceWM']
Fold: 103/103, Class 0 samples: 62, Class 1 samples: 40
Selected features: ['LVermilionH', 'NoseVol', 'NasoFacialA', 'NoseW', 'FaceWL']

Figure 1

4. The Test Performance

- Accuracy, Sensitivity, Specificity, AUC are shown in Figure 2

```
Test Performance
Confusion Matrix:
[[56  6]
 [ 7 34]]
Accuracy=0.874, Sensitivity=0.829, Specificity=0.903, AUC=0.906
```

Figure 2

- The ROC curve is shown in Figure 3

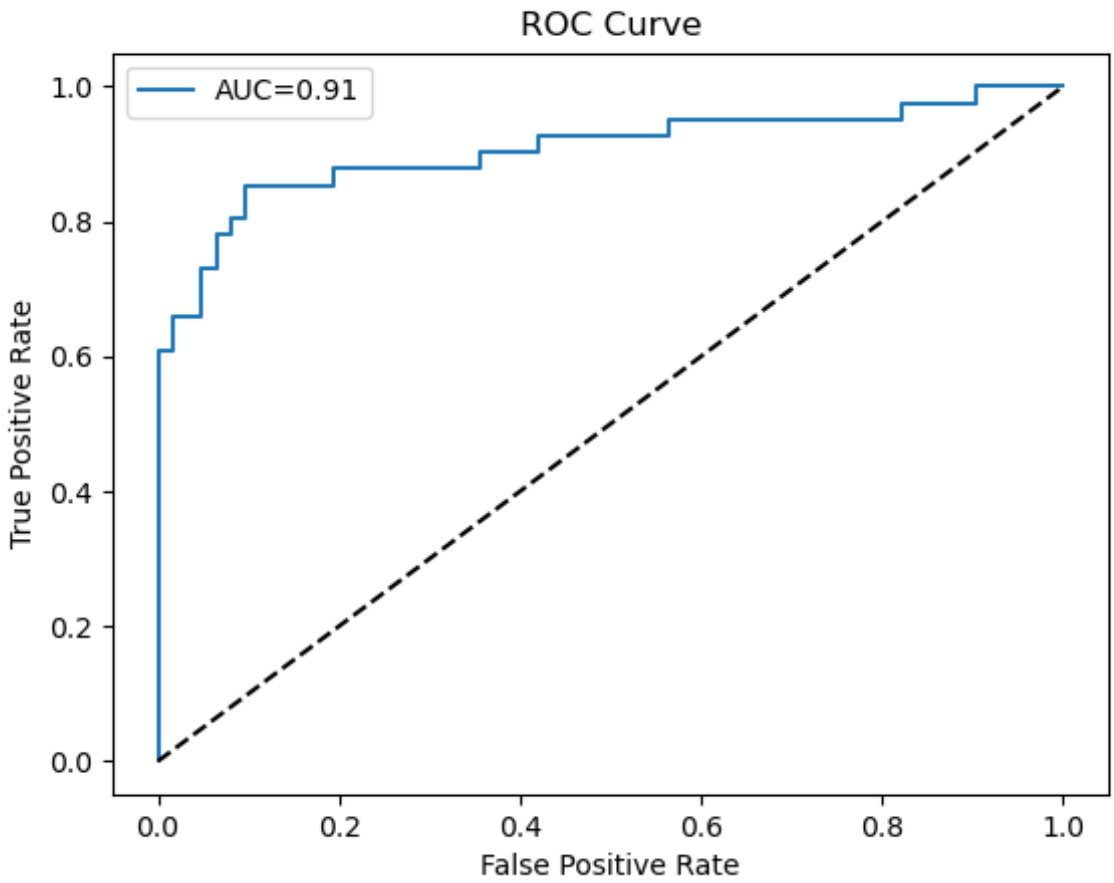


Figure 3

5. Top-2 Most Frequently Selected Features and Bivariate Gaussian Bayesian Model Visualization

- Top-2 Most Frequently Selected Features use `counts = Counter()` and `counts.most_common(2)` to find , shown in Figure 4

```
Top-2 features: [('NoseVol', 92), ('LVermilionH', 87)]
```

Figure 4

- Bivariate Gaussian Bayesian Model Visualization is shown in Figure 5
  - Scatter plot: class 0 as circles o(gray), class 1 as plus signs +(green)
  - Contour maps: equal-probability contours of each class distribution
  - Decision boundary: the curve where $\Delta(\mathbf{x}) = 0$

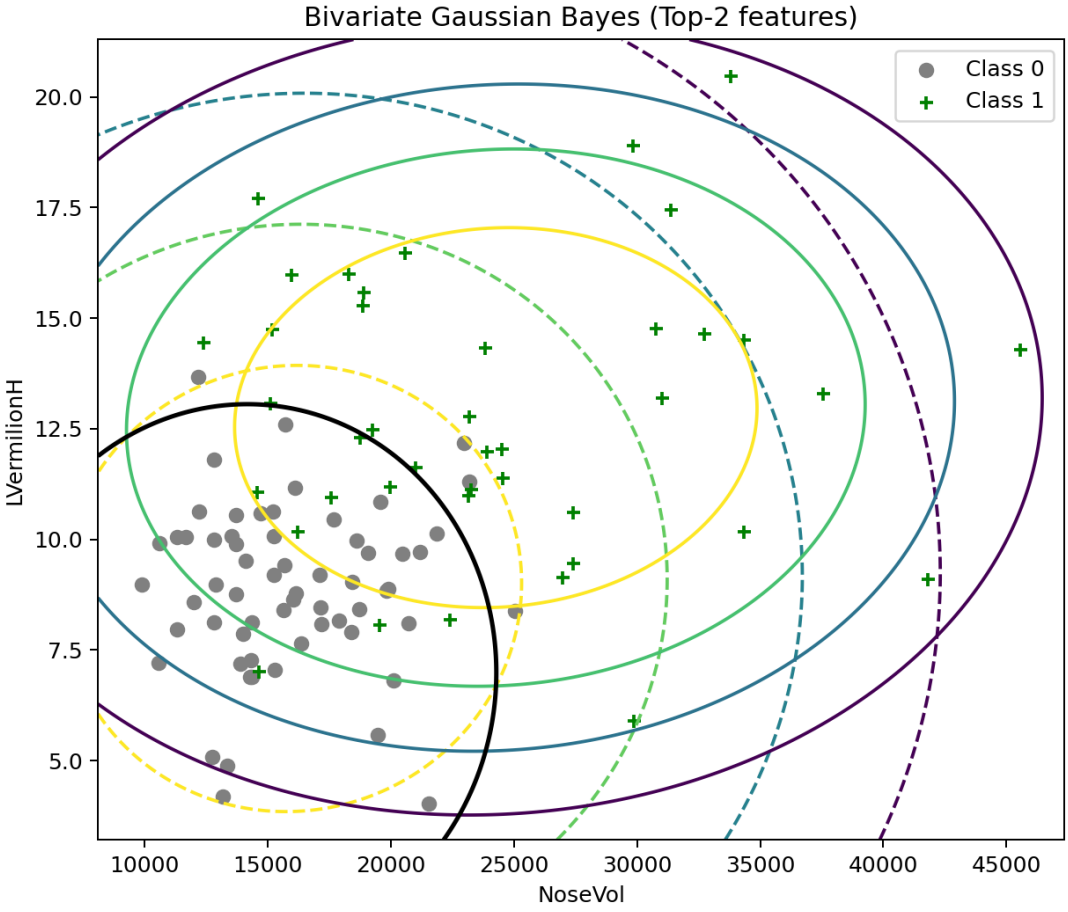
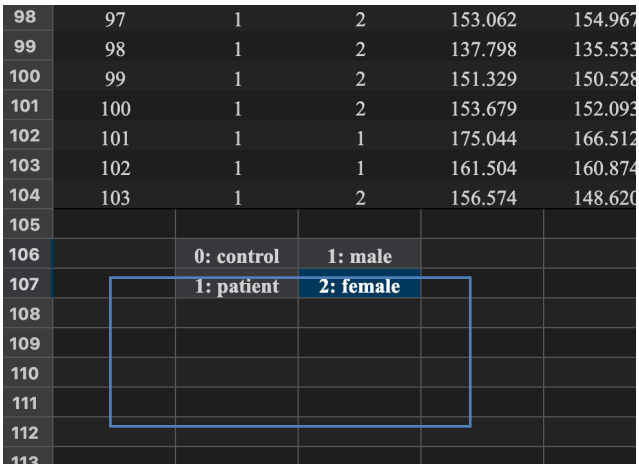


Figure 5

II. Code

Set up the environment `environment.yml` and run `1.py` shown in Figure 7, making sure to remove unnecessary information from `AcromegalyFeatureSet.xlsx` (eg. Figure 6)



98	97	1	2	153.062	154.967
99	98	1	2	137.798	135.533
100	99	1	2	151.329	150.528
101	100	1	2	153.679	152.093
102	101	1	1	175.044	166.512
103	102	1	1	161.504	160.874
104	103	1	2	156.574	148.620
105					
106		0: control	1: male		
107		1: patient	2: female		
108					
109					
110					
111					
112					

Figure 6. The blue rectangle must be deleted

```python
import numpy as np
import pandas as pd
from sklearn.metrics import roc_curve, auc, confusion_matrix
import matplotlib.pyplot as plt
from collections import Counter
from sklearn.model_selection import StratifiedKFold
from sklearn.metrics import roc_auc_score

''' Multivariate Gaussian Bayesian classifier '''
def Multivariate_Gaussian_Distribution_log_likelihood(X, mu, cov):
    X = np.atleast_2d(X)
    d = X.shape[1]
    L = np.linalg.cholesky(cov)
    Z = np.linalg.solve(L, (X - mu).T)
    quad = np.sum(Z*Z, axis=0)
    logdet = 2.0 * np.sum(np.log(np.diag(L)))
    log_likelihood = -0.5 * (d*np.log(2*np.pi) + logdet + quad)
    return log_likelihood

def Bayesian_decision_classifier(X, mu0, mu1, cov0, cov1, p0, p1, eps=1e-15):
    ll0 = Multivariate_Gaussian_Distribution_log_likelihood(X, mu0, cov0) + np.log(p0 + eps)  # (n,)
    ll1 = Multivariate_Gaussian_Distribution_log_likelihood(X, mu1, cov1) + np.log(p1 + eps)  # (n,)
    delta = ll1 - ll0
    m = np.maximum(ll0, ll1)
    num1 = np.exp(ll1 - m)
    den  = np.exp(ll0 - m) + num1
    posterior_1 = num1 / den
    # If single sample, return scalars
    if posterior_1.shape[0] == 1:
        return float(delta[0]), float(posterior_1[0])
    return delta, posterior_1

''' MLE Estimator for Gaussian parameters '''
def MLE_Estimater(X, y, ridge=1e-6):
    X0, X1 = X[y == 0], X[y == 1]
    mu0, mu1 = X0.mean(0), X1.mean(0)
    cov0 = np.atleast_2d(np.cov(X0, rowvar=False)) + np.eye(X.shape[1]) * ridge
    cov1 = np.atleast_2d(np.cov(X1, rowvar=False)) + np.eye(X.shape[1]) * ridge
    p0, p1 = len(X0)/len(X), len(X1)/len(X)
    return mu0, mu1, cov0, cov1, p0, p1

''' Cross-Validation AUC Score '''
def cv_AUC_score(X_train_sub, y_train_sub, max_splits=5, random_state=42):
    n_pos = int(np.sum(y_train_sub == 1))
    n_neg = int(np.sum(y_train_sub == 0))
    n_splits = max(2, min(max_splits, n_pos, n_neg))
    if n_pos == 0 or n_neg == 0 or n_splits < 2:
        return 0.5

    skf = StratifiedKFold(n_splits=n_splits, shuffle=True, random_state=random_state)
    y_all, s_all = [], []
    for tr_idx, va_idx in skf.split(X_train_sub, y_train_sub):
        Xtr, Xva = X_train_sub[tr_idx], X_train_sub[va_idx]
        ytr, yva = y_train_sub[tr_idx], y_train_sub[va_idx]
        mu0, mu1, cov0, cov1, p0, p1 = MLE_Estimater(Xtr, ytr, ridge=1e-6)
        _, s = Bayesian_decision_classifier(Xva, mu0, mu1, cov0, cov1, p0, p1)
        y_all.append(yva); s_all.append(s)
    y_all = np.concatenate(y_all); s_all = np.concatenate(s_all)
    try:
        return roc_auc_score(y_all, s_all)
    except Exception:
        return 0.5
```

```python
''' Forward Feature Selection based on CV-AUC '''
def Forward_Selection_AUC(X_train, y_train, max_features):
    remaining = list(range(X_train.shape[1]))
    selected = []
    best_score = -np.inf

    while remaining and len(selected) < max_features:
        candidates = []
        for f in remaining:
            trial = selected + [f]
            score = cv_AUC_score(X_train[:, trial], y_train)
            candidates.append((score, f))
        new_score, new_f = max(candidates, key=lambda x: x[0])

        if  (new_score > best_score + 1e-9):
            best_score = new_score
            selected.append(new_f)
            remaining.remove(new_f)
    return selected

''' Load Data '''
data = pd.read_excel('AcromegalyFeatureSet.xlsx')
data.rename(columns=lambda s: s.strip() if isinstance(s, str) else s, inplace=True)
X = data.drop(columns=['SeqNum','Gender', 'GroundTruth']).values
y = data['GroundTruth'].values
feature_names = data.drop(columns=['SeqNum','Gender', 'GroundTruth']).columns
n = len(y)

''' Leave-One-Out Cross Validation '''
posts1 = []
selected_features = []
deltas = []
print(f"Leave-One-Out Cross Validation")
for i in range(n):
    X_train = np.delete(X, i, axis=0)
    y_train = np.delete(y, i)
    X_test = X[i]

    n0 = int(np.sum(y_train == 0))
    n1 = int(np.sum(y_train == 1))
    print(f"------------------------------------------------------------")
    print(f"Fold: {i+1}/{n}, Class 0 samples: {n0}, Class 1 samples: {n1}")

    max_features = min(max(1, min(n0, n1) - 1), X_train.shape[1])

    selected = Forward_Selection_AUC(X_train, y_train, max_features)

    selected_features.append([feature_names[j] for j in selected])
    print(f"Selected features: {[feature_names[j] for j in selected]}")
    X_train_sel = X_train[:, selected]
    X_test_sel = X_test[selected]

    # mu0, mu1 = mean vectors for class 0 and 1
    mu0 = X_train_sel[y_train == 0].mean(axis=0)
    mu1 = X_train_sel[y_train == 1].mean(axis=0)

    # cov0, cov1 = covariance matrices for class 0 and 1
    cov0 = np.atleast_2d(np.cov(X_train_sel[y_train == 0], rowvar=False)) + np.eye(X_train_sel.shape[1]) * 1e-6
    cov1 = np.atleast_2d(np.cov(X_train_sel[y_train == 1], rowvar=False)) + np.eye(X_train_sel.shape[1]) * 1e-6

    # p0 = prior for class 0, p1 = prior for class 1
    p0, p1 = np.mean(y_train == 0), np.mean(y_train == 1)

    delta, posterior1 = Bayesian_decision_classifier(X_test_sel, mu0, mu1, cov0, cov1, p0, p1)
    posts1.append(posterior1)
    deltas.append(delta)
posts1 = np.array(posts1)
deltas = np.array(deltas)

''' Performance '''
fpr, tpr, _ = roc_curve(y, posts1)
roc_auc = auc(fpr, tpr)
preds1 = (deltas >= 0).astype(int)
cm = confusion_matrix(y, preds1)
TN, FP, FN, TP = cm.ravel()
acc = (TP + TN) / np.sum(cm)
sen = TP / (TP + FN)
spe = TN / (TN + FP)
print(f"------------------------------------------------------------")
print("Test Performance")
print(f"Confusion Matrix:\n{cm}")
print(f"Accuracy={acc:.3f}, Sensitivity={sen:.3f}, Specificity={spe:.3f}, AUC={roc_auc:.3f}")

# ROC curve plot
plt.plot(fpr, tpr, label=f"AUC={roc_auc:.2f}")
plt.plot([0, 1], [0, 1], 'k--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.legend()
plt.title('ROC Curve')
plt.savefig('roc_curve.png')

''' Bivariate Gaussian Bayes classifier with top-2 features '''
counts = Counter([f for fs in selected_features for f in fs])
print('Top-2 features:', counts.most_common(2))

Top2_features = [f for f, _ in counts.most_common(2)]

Top2_X = data[Top2_features].values.astype(float)
Top2_y = data['GroundTruth'].values.astype(int)

Top2_X0, Top2_X1 = Top2_X[Top2_y==0], Top2_X[Top2_y==1]
Top2_mu0, Top2_mu1 = Top2_X0.mean(0), Top2_X1.mean(0)
Top2_cov0 = np.cov(Top2_X0, rowvar=False) + np.eye(Top2_X0.shape[1]) * 1e-6
Top2_cov1 = np.cov(Top2_X1, rowvar=False) + np.eye(Top2_X1.shape[1]) * 1e-6
Top2_p0, Top2_p1 = len(Top2_X0)/len(Top2_X), len(Top2_X1)/len(Top2_X)
```

```
# grid points
m, M = Top2_X.min(0), Top2_X.max(0); pad = 0.05*(M-m)
xs = np.linspace(m[0]-pad[0], M[0]+pad[0], 300)
ys = np.linspace(m[1]-pad[1], M[1]+pad[1], 300)
xx, yy = np.meshgrid(xs, ys)
pts = np.c_[xx.ravel(), yy.ravel()]

# log-likelihoods and decision boundary
LL0 = Multivariate_Gaussian_Distribution_log_likelihood(pts, Top2_mu0, Top2_cov0).reshape(xx.shape)
LL1 = Multivariate_Gaussian_Distribution_log_likelihood(pts, Top2_mu1, Top2_cov1).reshape(xx.shape)
g0 = LL0 + np.log(Top2_p0)
g1 = LL1 + np.log(Top2_p1)
Top2_delta = g1 - g0 # decision boundary: delta=0

# plot
plt.figure(figsize=(7,6))
plt.scatter(Top2_X0[:,0], Top2_X0[:,1], marker='o', label='Class 0', color = 'gray')
plt.scatter(Top2_X1[:,0], Top2_X1[:,1], marker='+', label='Class 1', color = 'green')
levels0 = np.unique(np.sort(np.percentile(LL0, [20, 40, 60, 80])))
levels1 = np.unique(np.sort(np.percentile(LL1, [20, 40, 60, 80])))

plt.contour(xx, yy, LL0, levels=levels0, linestyles='dashed')
plt.contour(xx, yy, LL1, levels=levels1, linestyles='solid')
  # The decision boundary
  plt.contour(xx, yy, Top2_delta, levels=[0.0], linewidths=2, colors='black')

  plt.xlabel(Top2_features[0]); plt.ylabel(Top2_features[1])
  plt.title('Bivariate Gaussian Bayes (Top-2 features)')
  plt.legend(); plt.tight_layout()
  plt.savefig('bivariate_bayes_qda.png', dpi=180)
```

Figure 7

Problem 2. Proof of Bayesian estimator.

Suppose $x^t \sim N(\theta, \sigma^2)$ and $\theta \sim N(u_0, \sigma_0^2)$, where $u_0, \sigma_0^2, \sigma^2$ are known. That is

$$p(X|\theta) = \frac{1}{(2\pi)^{N/2}\sigma^N} \exp\left[-\frac{\sum_t (x^t - \theta)^2}{2\sigma^2}\right]$$

$$p(\theta) = \frac{1}{\sqrt{2\pi}\sigma_0} \exp\left[-\frac{(\theta - \mu_0)^2}{2\sigma_0^2}\right]$$

Please show that

$$E[\theta|X] = \frac{N/\sigma^2}{N/\sigma^2 + 1/\sigma_0^2} m + \frac{1/\sigma_0^2}{N/\sigma^2 + 1/\sigma_0^2} \mu_0$$

where m is the maximum likelihood estimator of the sample mean.

suppose Data. $x^t \sim N(\theta, \sigma^2)$ $t \in \{1, \cdots, N\}$
Prior distribution $\theta \sim N(\mu_0, \sigma_0^2)$
$\Rightarrow p(X|\theta) = \frac{1}{(2\pi)^{\frac{N}{2}}\sigma^N} \exp\left[-\frac{\sum_t (x^t-\theta)^2}{2\sigma^2}\right]$
$p(\theta) = \frac{1}{\sqrt{2\pi}\sigma_0} \exp\left[-\frac{(\theta-\mu_0)^2}{2\sigma_0^2}\right]$

由 Bayes' Rule 得知 $p(\theta|X) \propto p(X|\theta)\, p(\theta)$

$\Rightarrow p(\theta|X) \propto \exp\left[-\frac{1}{2\sigma^2}\sum_t (x^t-\theta)^2 - \frac{1}{2\sigma_0^2}(\theta-\mu_0)^2\right]$ $\because \mu_0, \sigma_0^2, \sigma^2$ are know

$\sum_t (x^t-\theta)^2 = \sum_t (x^t)^2 - 2\theta\sum_t x^t + N\theta^2$, $(\theta-\mu_0)^2 = \theta^2 - 2\theta\mu_0 + \mu_0^2$

$\Rightarrow -\frac{1}{2\sigma^2}\sum_t (x^t-\theta)^2 - \frac{1}{2\sigma_0^2}(\theta-\mu_0)^2 = -\frac{1}{2}\left[\frac{N}{\sigma^2}\theta^2 - 2\frac{\sum_t x^t}{\sigma^2}\theta + \frac{1}{\sigma_0^2}\theta^2 - 2\frac{\mu_0}{\sigma_0^2}\theta\right] + \underbrace{\left(-\frac{\mu_0^2}{2\sigma_0^2} - \frac{1}{2\sigma^2}\sum_t (x^t)^2\right)}_{\text{constant}}$

set $A = \frac{N}{\sigma^2} + \frac{1}{\sigma_0^2}$, $B = \frac{\sum_t x^t}{\sigma^2} + \frac{\mu_0}{\sigma_0^2}$

$\Rightarrow p(\theta|X) \propto \exp\left[-\frac{1}{2}(A\theta^2 - 2B\theta)\right] = \exp\left[-\frac{A}{2}(\theta^2 - 2\cdot\frac{B}{A}\theta)\right] \propto \exp\left[-\frac{A}{2}(\theta^2 - 2\cdot\frac{B}{A}\theta + \underbrace{(\frac{B}{A})^2}_{\text{constant}})\right]$

$= \exp\left[-\frac{\theta}{2}(\theta - \frac{B}{A})^2\right]$

$\Rightarrow p(\theta|X) = N(\text{mean} = \frac{B}{A}, \text{varience} = \frac{1}{A})$

$\Rightarrow E(\theta|X) = \frac{B}{A} = \frac{\frac{\sum_t x^t}{\sigma^2} + \frac{\mu_0}{\sigma_0^2}}{\frac{N}{\sigma^2} + \frac{1}{\sigma_0^2}}$

$\because m = \frac{1}{N}\sum_t x^t$ the maximum likelihood of sample mean $\Leftrightarrow \sum_t x^t = Nm$

$\Rightarrow E(\theta|X) = \frac{\frac{Nm}{\sigma^2} + \frac{\mu_0}{\sigma_0^2}}{\frac{N}{\sigma^2} + \frac{1}{\sigma_0^2}} = \frac{N/\sigma^2}{N/\sigma^2 + 1/\sigma_0^2} m + \frac{1/\sigma_0^2}{N/\sigma^2 + 1/\sigma_0^2}\mu_0$ 得證 #