# NYCU Introduction to Machine Learning, Homework 2

**Deadline: Nov. 1, 23:59**

## **Part. 1, Coding (60%)**:

In this coding assignment, you are required to implement Fisher's linear discriminant by using only NumPy, then train your model on the provided dataset, and evaluate the performance on testing data. Find the sample code and data on the GitHub page https://github.com/NCTU-VRDL/CS_CS20024/tree/main/HW2

**Please note that only NumPy can be used to implement your model, you will get 0 point by calling sklearn.discriminant_analysis.LinearDiscriminantAnalysis.**

1. (5%) Compute the mean vectors $m_i$ (i=1, 2) of each 2 classes on **training data**

```
mean vector of class 1: [ 0.99253136 -0.99115481] mean vector of class 2: [-0.9888012   1.00522778]
```

2. (5%) Compute the within-class scatter matrix $S_W$ on **training data**

```
Within-class scatter matrix SW: [[ 4337.38546493 -1795.55656547]
 [-1795.55656547  2834.75834886]]
```

3. (5%) Compute the between-class scatter matrix $S_B$ on **training data**

```
Between-class scatter matrix SB: [[ 3.92567873 -3.95549783]
 [-3.95549783  3.98554344]]
```

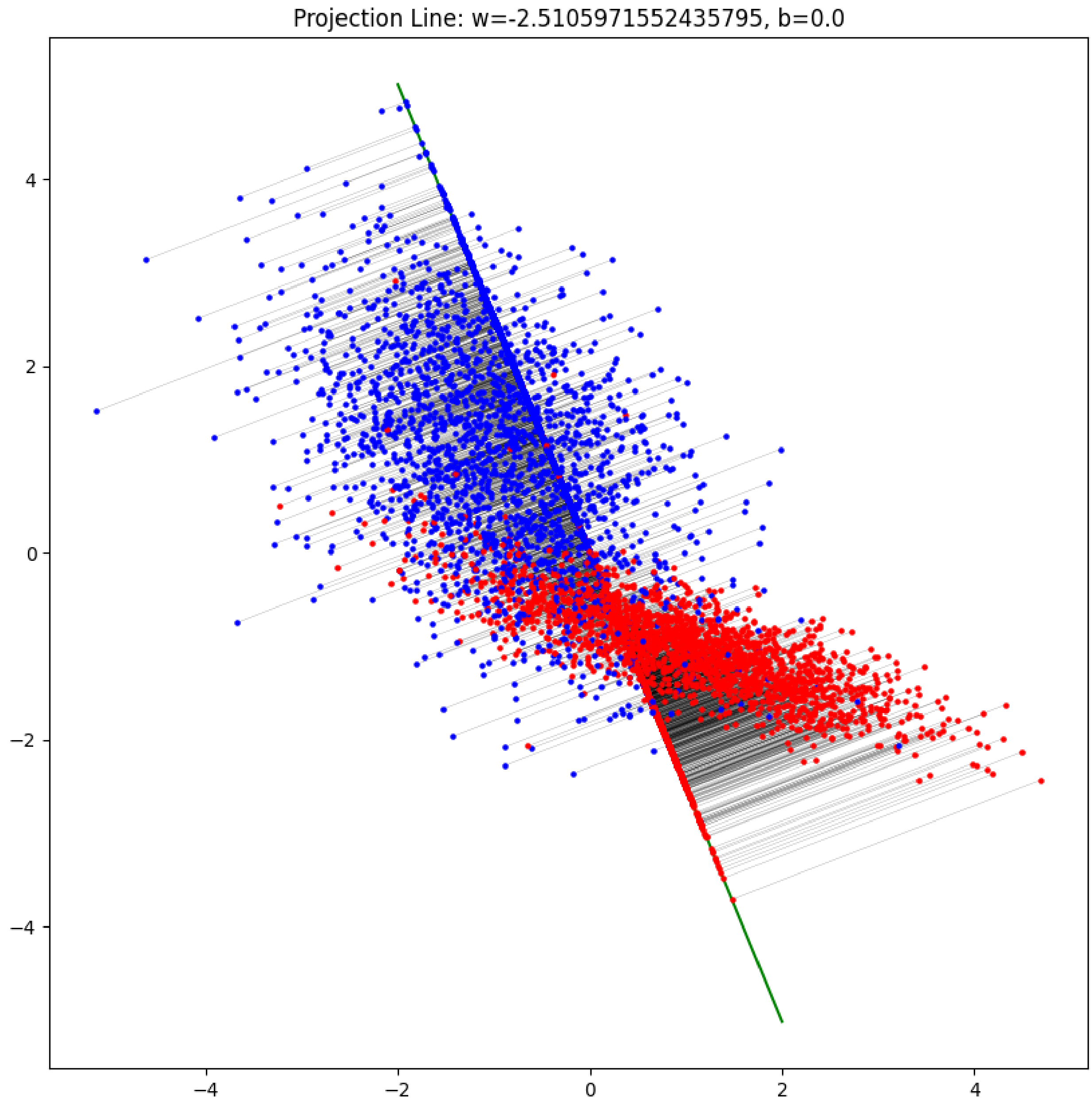
4. (5%) Compute the Fisher's linear discriminant $w$ on **training data**

```
Fisher's linear discriminant: [[-0.37003809]
[ 0.92901658]]
```

5. (20%) Project the **testing data** by Fisher's linear discriminant to get the class prediction by K-Nearest-Neighbor rule and report the accuracy score on **testing data** with K values from 1 to 5 (you should get accuracy over **0.88**)

```
K=1: Accuracy of test-set 0.8488
K=2: Accuracy of test-set 0.8704
K=3: Accuracy of test-set 0.8792
K=4: Accuracy of test-set 0.8824
K=5: Accuracy of test-set 0.8912
```

6. (20%) Plot the **1) best projection line** on the **<u>training data</u>** and <u>show the slope and intercept on the title</u> *(you can choose any value of* ***intercept*** *for better visualization)*
**2) colorize the data** with each class **3)** project all data points on your projection line.
Your result should look like the below image (This image is for reference, not the answer)




## Part. 2, Questions (40%):

**Please write/type by yourself. DO NOT screenshot the solution from others.**

(10%) 1. What's the difference between the Principle Component Analysis and Fisher's Linear Discriminant?

Principal Component Analysis: 將高維度數據透過投影降低維度，找到包含數據最大差異性的主成分方向。

Fisher's Linear Discriminant: PCA的延伸，以分類為目標。因此考慮數據類別資訊，希望投影後的資料組內分散量越小越好、組間分散量越大越好。

(10%) 2. Please explain in detail how to extend the 2-class FLD into multi-class FLD (the number of classes is greater than two).

有K個class時，計算J(w)時需要的Sb和Sw需考慮K個class:

$$\boxed{S_W = \sum_{k=1}^{K} S_k} \rightarrow \text{K個class的}S_k\text{相加}$$

$$S_k = \sum_{n \in C_k} (x_n - m_k)(x_n - m_k)^T$$

↳ class k 中的資料分散量

$$m_k = \frac{1}{N_k} \sum_{n \in C_k} x_n \rightarrow \text{class k平均}$$

$$\boxed{S_B = \sum_{k=1}^{K} N_k (m_k - m)(m_k - m)^T} \rightarrow \text{K個class之間的資料分散量}$$

→ class k 和平均差距 * class k 資料數

$$m = \frac{1}{N} \sum_{n=1}^{N} x_n \rightarrow \text{所有資料平均}$$

(6%) 3. By making use of Eq (1) ~ Eq (5), show that the Fisher criterion Eq (6) can be written in the form Eq (7).

$$y = \mathbf{w}^{\mathrm{T}}\mathbf{x} \qquad \text{Eq (1)}$$

$$\mathbf{m}_1 = \frac{1}{N_1}\sum_{n \in \mathcal{C}_1} \mathbf{x}_n \qquad \mathbf{m}_2 = \frac{1}{N_2}\sum_{n \in \mathcal{C}_2} \mathbf{x}_n \qquad \text{Eq (2)}$$

$$m_2 - m_1 = \mathbf{w}^{\mathrm{T}}(\mathbf{m}_2 - \mathbf{m}_1) \qquad \text{Eq (3)}$$

$$m_k = \mathbf{w}^{\mathrm{T}}\mathbf{m}_k \qquad \text{Eq (4)}$$

$$s_k^2 = \sum_{n \in \mathcal{C}_k} (y_n - m_k)^2 \qquad \text{Eq (5)}$$

$$J(\mathbf{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2} \qquad \text{Eq (6)}$$

$$J(\mathbf{w}) = \frac{\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{B}}\mathbf{w}}{\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{W}}\mathbf{w}} \qquad \text{Eq (7)}$$

證 $\dfrac{w^T S_B w}{w^T S_W w} = \dfrac{(m_2 - m_1)^2}{s_1^2 + s_2^2}$ :

$\underline{w^T S_B w} = w^T(m_2 - m_1)(m_2 - m_1)^T w$

$m_2 - m_1 = w^T(m_2 - m_1) \Rightarrow (m_2 - m_1)^2 = w^T(m_2 - m_1)(m_2 - m_1)^T w$

$\Rightarrow \underline{w^T S_B w = (m_2 - m_1)^2}$ #

$s_k^2 = \sum_{y \in C_k} (\underline{\overset{=w^T x_n}{y_n}} - \underline{\overset{=w^T m_k}{m_k}})^2 = \sum_{k \in C_k} (w^T x_n - w^T m_k)^2$

$= \sum_{k \in C_k} w^T (x_n - m_k)(x_n - m_k) w$

$k = 1, 2$

$\Rightarrow \underline{s_1^2 + s_2^2}$

$= w^T \Big( \underbrace{\sum_{k \in C_1} (x_n - m_1)(x_n - m_1)^T + \sum_{k \in C_2} (x_n - m_2)(x_n - m_2)^T}_{S_w} \Big) w$

$= \underline{w^T S_w w}$ #

$\Rightarrow$ 得證 $\dfrac{w^T S_B w}{w^T S_W w} = \dfrac{(m_2 - m_1)^2}{s_1^2 + s_2^2}$

(7%) 4. Show the derivative of the error function Eq (8) with respect to the activation $a_k$ for an output unit having a logistic sigmoid activation function satisfies Eq (9).

$$E(\mathbf{w}) = -\sum_{n=1}^{N} \{t_n \ln y_n + (1 - t_n) \ln(1 - y_n)\} \quad \text{Eq (8)}$$

$$\frac{\partial E}{\partial a_k} = y_k - t_k \quad \text{Eq (9)}$$

$y_k = \sigma(a_k)$ , $\sigma$ : logistic sigmoid function

$$\boxed{\frac{\partial E(w)}{\partial a_k}} = -t_k \frac{1}{y_k}[y_k(1-y_k)] + (1-t_k)\frac{1}{1-y_k}[y_k(1-y_k)]$$

$$= [y_k(1-y_k)]\left[\frac{1-t_k}{1-y_k} - \frac{t_k}{y_k}\right]$$

$$= (1-t_k)y_k - t_k(1-y_k) = \underline{y_k - t_k} \quad \#$$

(7%) 5. Show that maximizing likelihood for a multiclass neural network model in which the network outputs have the interpretation $y_k(x, w) = p(t_k = 1 \mid x)$ is equivalent to the minimization of the cross-entropy error function Eq (10).

$$E(\mathbf{w}) = -\sum_{n=1}^{N} \sum_{k=1}^{K} t_{kn} \ln y_k(\mathbf{x}_n, \mathbf{w}) \quad \text{Eq (10)}$$

$$y_k(x, w) = p(t_k = 1 \mid x) = y_{nk}$$

$$E(w) = -\ln \prod^{N} p(t \mid x_n, w)$$

$$= -\ln \prod_{n=1}^{N} \prod_{k=1}^{K} y_k(x_n, w)^{t_{nk}} \left[1 - y_k(x_n, w)\right]^{1 - t_{nk}}$$

$$= -\sum_{n=1}^{N} \sum_{k=1}^{K} \ln \left\{ y_k(x_n, w)^{t_{nk}} \left[1 - y_k(x_n, w)\right]^{1 - t_{nk}} \right\}$$

$$= -\sum_{n=1}^{N} \sum_{k=1}^{K} \ln \left[ y_{nk}^{t_{nk}} (1 - y_{nk})^{1 - t_{nk}} \right]$$

$$= -\sum_{n=1}^{N} \sum_{k=1}^{K} \left\{ t_{nk} \ln y_{nk} + (1 - t_{nk}) \ln (1 - y_{nk}) \right\}$$