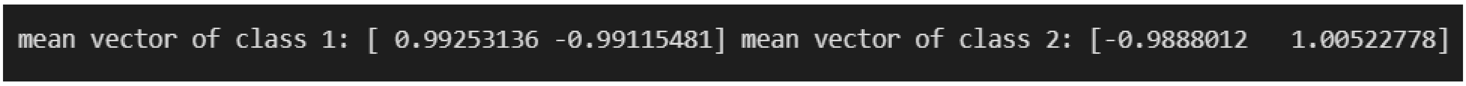
HW2

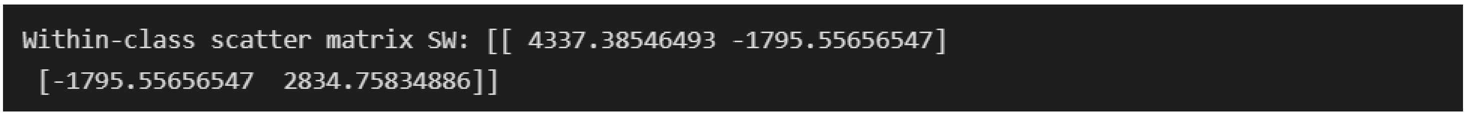
Part. 1, Coding (60%):

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



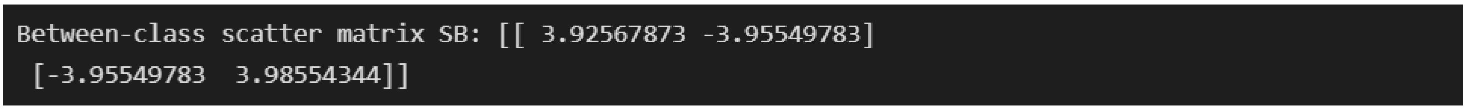
```
mean vector of class 1: [ 0.99253136 -0.99115481] mean vector of class 2: [-0.9888012   1.00522778]
```

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



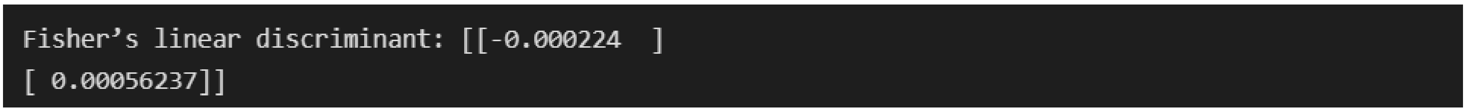
```
Within-class scatter matrix SW: [[ 4337.38546493 -1795.55656547]
 [-1795.55656547  2834.75834886]]
```

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



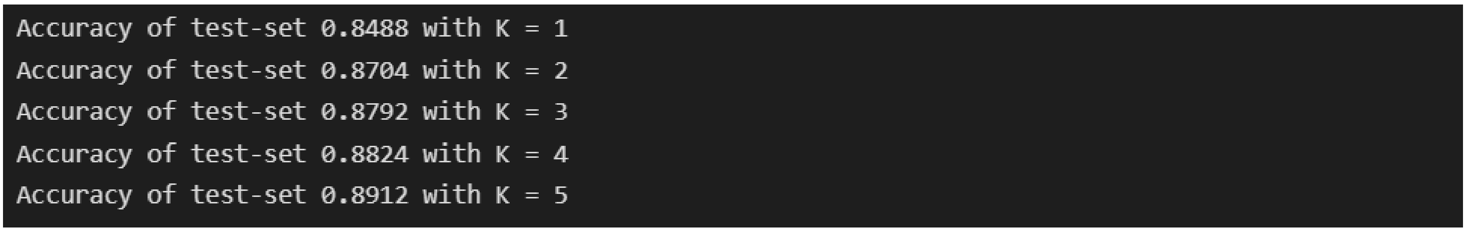
```
Between-class scatter matrix SB: [[ 3.92567873 -3.95549783]
 [-3.95549783  3.98554344]]
```

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



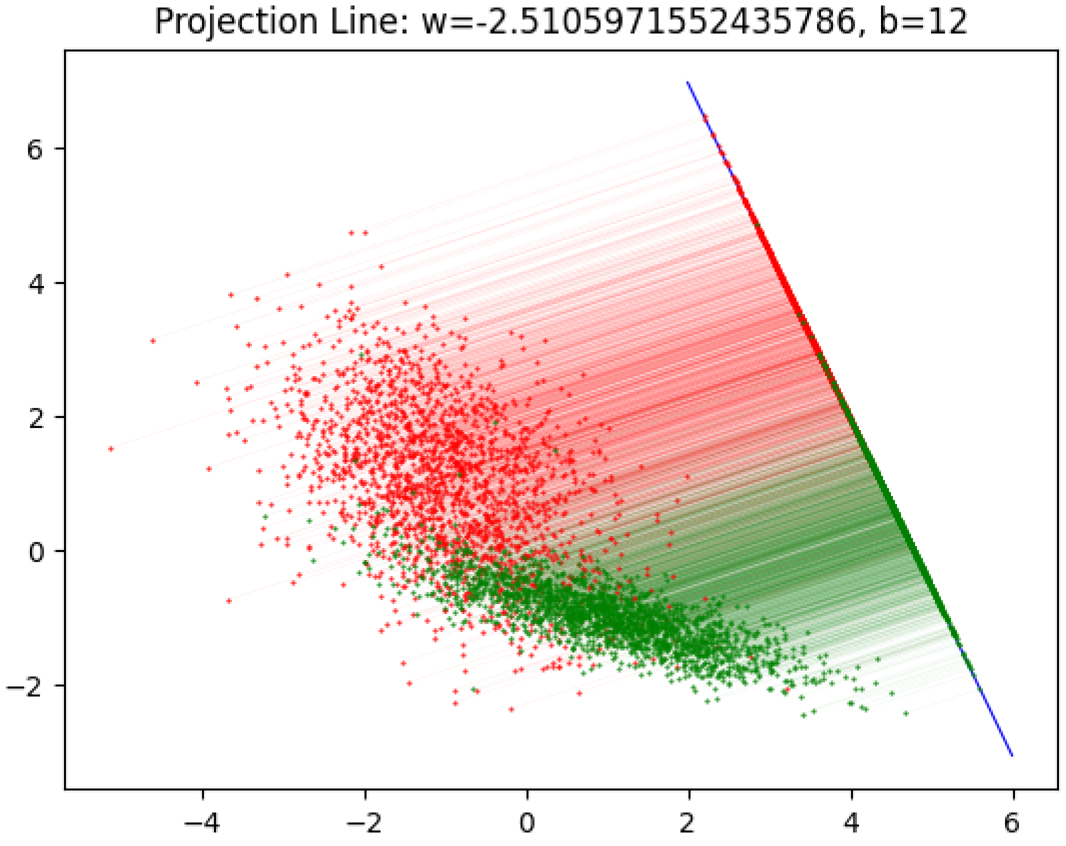
```
Fisher's linear discriminant: [[-0.000224  ]
[ 0.00056237]]
```

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)



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

6. (20%) Plot the 1) best projection line on the training data and show the slope and intercept on the title *(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%):

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

Principle Component Analysis 是非監督的降維方法，可以降低任意維度，選擇樣本點具有最大 variance 的方向進行降維。

Fisher’s Linear Discriminant 是有監督的降維方法，同時可以拿來做分類，但只能將數據降低一個維度，降維方向選擇不同類的 mean 離最遠且每類的 variance 最小。

(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).

根據 2-class FLD 的 S_B, S_W 的形式定義當有 k 個 class 時的 S_B, S_W 如下：

The within-class covariance matrix when $K \geq 2$:

$S_W = \Sigma_{k=1}^K S_k$, where $S_k = \Sigma_{n \in C_k}(x_n - m_k)(x_n - m_k)^T$ and $m_k = \frac{1}{N_k} x_n$

The extended between-class covariance matrix for $K > 2$:

$S_B = \Sigma_{k=1}^K N_k(m_k - m)(m_k - m)^T$, where $m = \frac{1}{N}\Sigma_{n=1}^N x_n$

一樣可以得出這個式子 $J(w) = \frac{w^T S_B w}{w^T S_w w}$

然後對此最佳化求 Maximize S_B, Minimize S_w 即為 multi-class FLD

(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).

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

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

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

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

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

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

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

$S_B = (m_2 - m_1)(m_2 - m_1)^T$

$S_W = \Sigma_{n \in C_1}(x_n - m_1)(x_n - m_1)^T + \Sigma_{n \in C_2}(x_n - m_2)(x_n - m_2)^T$

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

$= \frac{(w^T(m_2 - m_1))^2}{s_1^2 + s_2^2}$b by Eq(3)

$= \frac{(w^T(m_2 - m_1))(w^T(m_2 - m_1))}{s_1^2 + s_2^2}$

$= \frac{(w^T(m_2 - m_1))((m_2 - m_1)^T w)}{s_1^2 + s_2^2}$

$= \frac{w^T S_B w}{s_1^2 + s_2^2}$

$= \frac{w^T S_B w}{\Sigma_{n \in C_1}(y_n - m_1)^2 + \Sigma_{n \in C_2}(y_n - m_2)^2}$ by Eq(5)

$= \frac{w^T S_B w}{\Sigma_{n \in C_1}(w^T x_n - m_1)^2 + \Sigma_{n \in C_2}(w^T x_n - m_2)^2}$ by Eq(1)

$= \frac{w^T S_B w}{\Sigma_{n \in C_1}(w^T x_n - w^T m_1)^2 + \Sigma_{n \in C_2}(w^T x_n - w^T m_2)^2}$ by Eq(4)

$= \frac{w^T S_B w}{\Sigma_{n \in C_1}(w^T x_n - w^T m_1)(w^T x_n - w^T m_1) + \Sigma_{n \in C_2}(w^T x_n - w^T m_2)(w^T x_n - w^T m_2)}$

$= \frac{w^T S_B w}{\Sigma_{n \in C_1} w^T(x_n - m_1) w^T(x_n - m_1) + \Sigma_{n \in C_2} w^T(x_n - m_2) w^T(x_n - m_2)}$

$= \frac{w^T S_B w}{\Sigma_{n \in C_1} w^T(x_n - m_1)(x_n - m_1)^T w + \Sigma_{n \in C_2} w^T(x_n - m_2)(x_n - m_2)^T w}$

$= \frac{w^T S_B w}{w^T(\Sigma_{n \in C_1}(x_n - m_1)(x_n - m_1)^T + \Sigma_{n \in C_2}(y_n - m_2)(y_n - m_2)^T) w}$

$= \frac{w^T S_B w}{w^T S_W w}$ Eq(7)

(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).

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

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

Suppose $t_{n1} = t_n$, $t_{n2} = 1 - t_n$, $y_{n1} = y_n$, $y_{n2} = 1 - y_n$

Then $E(w) = -\Sigma_{n=1}^{N}\Sigma_{k=1}^{K} t_{nk} \ln y_{nk}$, where $K = 2$

However we have $\frac{\partial E}{\partial y_{nk}} = -\frac{t_{nk}}{y_{nk}}$ (eq. a)

According to $y_k = \frac{e^{a_k}}{\Sigma_j e^{a_j}}$

We have $\frac{\partial y_k}{\partial a_j} = y_k(I_{kj} - y_j)$, where $I_{kj} = \{1, j = k; 0, \text{otherwise}.$ (eq. b)

Given eq. a and eq. b, we can compute

$\frac{\partial E}{\partial a_{nj}} = \Sigma_{k=1}^{K} \frac{\partial E}{\partial y_{nk}} \frac{\partial y_{nk}}{\partial a_{nj}} = -\Sigma_{k=1}^{K} \frac{t_{nk}}{y_{nk}} y_{nk}(I_{kj} - y_{nj}) = -\Sigma_{k=1}^{K} t_{nk}(I_{kj} - y_{nj}) = -t_{nj} + \Sigma_{k=1}^{K} t_{nk} y_{nj} = y_{nj} - t_{nj}$. (eq. c)

Since eq. c, we have Eq (9)

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

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

$$\begin{aligned}
p(T|w_1, \ldots, w_k) &= \prod_{n=1}^{N} \prod_{k=1}^{K} y_k(x_n, w)^{t_{kn}} \\
E(w) &= -\ln\left(p(T|w_1, \ldots, w_k)\right) \\
&= -\ln\left(\prod_{n=1}^{N} \prod_{k=1}^{K} y_k(x_n, w)^{t_{kn}}\right) \\
&= -\sum_{n=1}^{N} \sum_{k=1}^{K} \ln y_k(x_n, w)^{t_{kn}} \\
&= -\sum_{n=1}^{N} \sum_{k=1}^{K} t_{kn} \ln y_k(x_n, w) \qquad \text{Eq(10)}
\end{aligned}$$