# NYCU Introduction to Machine Learning, Homework 2

[112550198], [簡嫚萱]

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

### (25%) Logistic Regression w/ Gradient Descent Method

1. (5%) Show the hyperparameters (learning rate and iteration, etc) that you used and the weights and intercept of your model.

```
class LogisticRegression:
    def __init__(self, learning_rate: float = 1e-2, num_iterations: int = 5000):
```

```
LR: Weights: [-0.72024348  0.10748414  0.93801098 -0.07279415  0.08648435],
```

```
Intercep: -2.45801749575814
```

2. (5%) Show the AUC of the classification results on the testing set.

```
AUC=0.8545
```

3. (15%) Show the accuracy score of your model on the testing set

```
LR: Accuracy=0.8333,
```

### (25%) Fisher Linear Discriminant, FLD

4. (5%) Show the mean vectors $m_i$ (i=0, 1) of each class, the within-class scatter matrix *Sw,* and the between-class scatter matrix *Sb* of the training set.

```
FLD: m0=[ 0.35994138 -0.04560139], m1=[0.32519126 0.04435118]
```

```
Sw=
[[41.93041055 15.7202037 ]
 [15.7202037  37.25186904]]
```

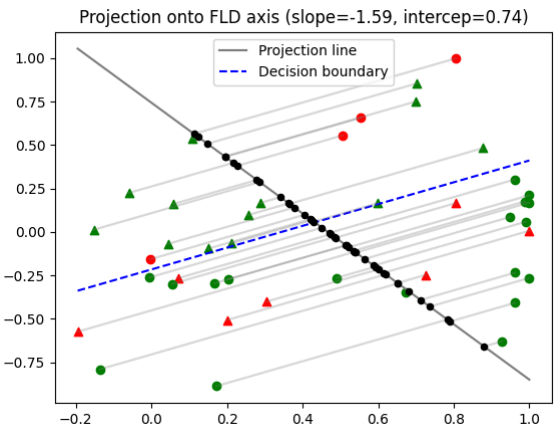
```
Sb=
[[ 0.00120757 -0.00312586]
 [-0.00312586  0.00809147]]
```

5. (5%) Show the Fisher's linear discriminant *w* of the training set.

```
W=
[-0.00205997  0.00328402]
```

6. (15%) Show the accuracy score on the testing set. Also, plot/obtain predictions for the testing set by measuring the distance between the projected value of the testing data and the projected means of the training data for the two classes (**Please check the slide for color, shape, and other plotting requirements**).

```
FLD: Accuracy=0.7381
```



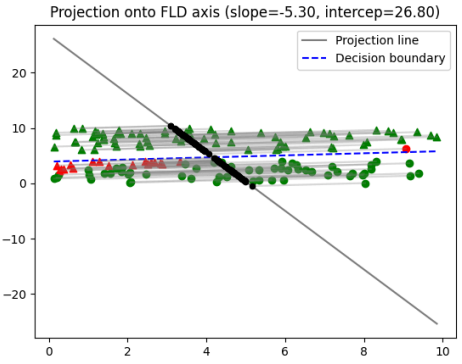
## (10%) Code Check and Verification

7. (10%) Lint the code and show the PyTest results.

```
============================= test session starts ==============================
platform linux -- Python 3.13.7, pytest-8.4.2, pluggy-1.6.0
rootdir: /home/michelle/Intro.ML/hw2
configfile: pyproject.toml
collected 2 items

test_main.py (395, 2) (395,)
iter 0: loss = 0.6931
iter 10000: loss = 0.0672
iter 20000: loss = 0.0493
2025-10-25 16:59:14.847 | INFO     | test_main:test_logistic_regression:35 - accu
racy=1.0000
.(395, 2) (395,)
2025-10-25 16:59:14.849 | INFO     | test_main:test_fld:45 - accuracy=0.8759
.

============================== 2 passed in 77.79s (0:01:17) ==============================
```




```
(hw2) michelle@LAPTOP-E7SHGCCE:~/Intro.ML/hw2$ flake8 main.py
(hw2) michelle@LAPTOP-E7SHGCCE:~/Intro.ML/hw2$
```

# Part. 2, Questions (40%):

1. (15%)

(⋆) Using (4.57) and (4.58), derive the result (4.65) for the posterior class probability in the two-class generative model with Gaussian densities, and verify the results (4.66) and (4.67) for the parameters $\mathbf{w}$ and $w_0$.

$$p(\mathcal{C}_1|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1) + p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)} = \frac{1}{1+\exp(-a)} = \sigma(a) \tag{4.57}$$

$$a = \ln \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)} \tag{4.58}$$

$$p(\mathcal{C}_1|\mathbf{x}) = \sigma(\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0) \tag{4.65}$$

$$\mathbf{w} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) \tag{4.66}$$

$$w_0 = -\frac{1}{2}\boldsymbol{\mu}_1^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_2^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_2 + \ln\frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)}. \tag{4.67}$$

1. ∵ Assume that the class-conditional densities are Gaussian and all classes share the same covariance matrix.

$$\therefore P(X|C_k) = \frac{1}{(2\pi)^{N/2}} \frac{1}{|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(X-\mu_k)^T\Sigma^{-1}(X-\mu_k)\right\}$$

and substitute $P(X|C_k)$ into 4.58

$$\Rightarrow a = \ln \frac{\frac{1}{(2\pi)^{N/2}}\frac{1}{|\Sigma|^{1/2}} \cdot \exp\left[-\frac{1}{2}(X-\mu_1)^T\Sigma^{-1}(X-\mu_1)\right]}{\frac{1}{(2\pi)^{N/2}}\frac{1}{|\Sigma|^{1/2}} \cdot \exp\left\{-\frac{1}{2}(X-\mu_2)^T\Sigma^{-1}(X-\mu_2)\right\}} + \ln\frac{P(C_1)}{P(C_2)} = -\frac{1}{2}\left[(X-\mu_1)^T\Sigma^{-1}(X-\mu_1) - (X-\mu_2)^T\Sigma^{-1}(X-\mu_2)\right] + \ln\frac{P(C_1)}{P(C_2)}$$

$$= -\frac{1}{2}\left(X^T\Sigma^{-1}X - 2\mu_1^T\Sigma^{-1}X + \mu_1^T\Sigma^{-1}\mu_1 - X^T\Sigma^{-1}X + 2\mu_2^T\Sigma^{-1}X - \mu_2^T\Sigma^{-1}\mu_2\right) + \ln\frac{P(C_1)}{P(C_2)}$$

$$= \mu_1^T\Sigma^{-1}X - \mu_2^T\Sigma^{-1}X - \frac{1}{2}\mu_1^T\Sigma^{-1}\mu_1 + \frac{1}{2}\mu_2^T\Sigma^{-1}\mu_2 + \ln\frac{P(C_1)}{P(C_2)}$$

$$= (\Sigma^{-1}\mu_1 - \Sigma^{-1}\mu_2)^T X - \frac{1}{2}\mu_1^T\Sigma^{-1}\mu_1 + \frac{1}{2}\mu_2^T\Sigma^{-1}\mu_2 + \ln\frac{P(C_1)}{P(C_2)}$$

$$\therefore \text{We get } P(C_1|X) = \sigma(w^TX + w_0), \text{ where } \begin{cases} w = \Sigma^{-1}(\mu_1 - \mu_2) \\ w_0 = -\frac{1}{2}\mu_1^T\Sigma^{-1}\mu_1 + \frac{1}{2}\mu_2^T\Sigma^{-1}\mu_2 + \ln\frac{P(C_1)}{P(C_2)} \end{cases} \quad \#$$

112550198 簡煖芮 2025. 10. 25

2. (10%)

(a) Give one real-world situation where you would prefer Logistic Regression (LR) over the Perceptron, and explain why.

(b) Is Logistic Regression actually used for regression (predicting a continuous value)? If not, state what task it really solves and why the name includes "regression."

(a) One example is pedestrian detection in autonomous driving. Logistic Regression is better for noisy datasets and when probability interpretation is important. Sensor data in this scenario often include noise, varying lighting conditions, or partial occlusion. The probability output is important because the system needs to decide whether to brake based on the likelihood that a pedestrian is present. Logistic Regression provides continuous confidence levels (e.g., 0.95), whereas the Perceptron's binary output can lead to overly extreme reactions.

(b) Generally no, but it can be used for solving regression-like problems if the predicted value represents a probability between 0 and 1 rather than exactly 0 or 1. The reason it is called "regression" is that the model estimates the parameters using a regression-like approach — it fits a linear function of the inputs ($w^T$x+b) but passes it through a nonlinear **logistic (sigmoid)** function to map it to a probability. In other words, it performs a **linear regression in the log-odds space**, not directly on the output variable.

3. (15%)

(a) Why is feature scaling (e.g., standardization or normalization) important in Logistic Regression? Explain two reasons.

(b) If feature scaling is not applied in Logistic Regression, list three problems that may occur. Briefly explain.

(a) Logistic Regression often uses gradient descent to estimate model parameters. One reason feature scaling is important is to **accelerate the convergence of gradient descent**. If the feature values differ greatly, the gradients along different directions will have very different scales, which can cause oscillations in some directions and slow down convergence. Another reason is to **avoid excessive influence of large-scale features on the model.** Without feature scaling, features with large scales may have an exaggerated influence on the model, potentially leading to incorrect judgments about feature importance.

(b) 1. **Slow and unstable gradient descent**:
   If features have very different scales, the gradients in different directions can vary greatly, which leads to oscillations or slow convergence when doing gradient descent.
2. **Model is overly sensitive to large-scale feature**:

The parameter in Logistic Regression is sensitive to the scale of the feature. Therefore, features with larger numerical values may have an exaggerated influence on the predictions, and this can cause incorrect estimations of feature importance.

3. **Difficulty in regularization**:
   Regularization penalizes coefficients based on their magnitude. Without feature scaling, large-scale features may be penalized differently, decreasing the effect of regularization.