NYCU Introduction to Machine Learning, Homework 2

112550077, 劉逢穎

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.

```
LR = LogisticRegression(
    learning_rate=1e-3,  # You can modify the parameters as you want
    num_iterations=2000,  # You can modify the parameters as you want
)
```

```
2025-10-25 02:05:14.683 | INFO     | __main__:main:215 - LR: Weights: [-0.02876961  0.0064896
9  0.02775944  0.02291609  0.01587524], Intercep: -0.09114388296871183
```

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

```
2025-10-25 02:05:14.683 | INFO     | __main__:main:216 - LR: Accuracy=0.8095, AUC=0.8455
```

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

```
2025-10-25 02:05:14.683 | INFO     | __main__:main:216 - LR: Accuracy=0.8095, AUC=0.8455
```

(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.

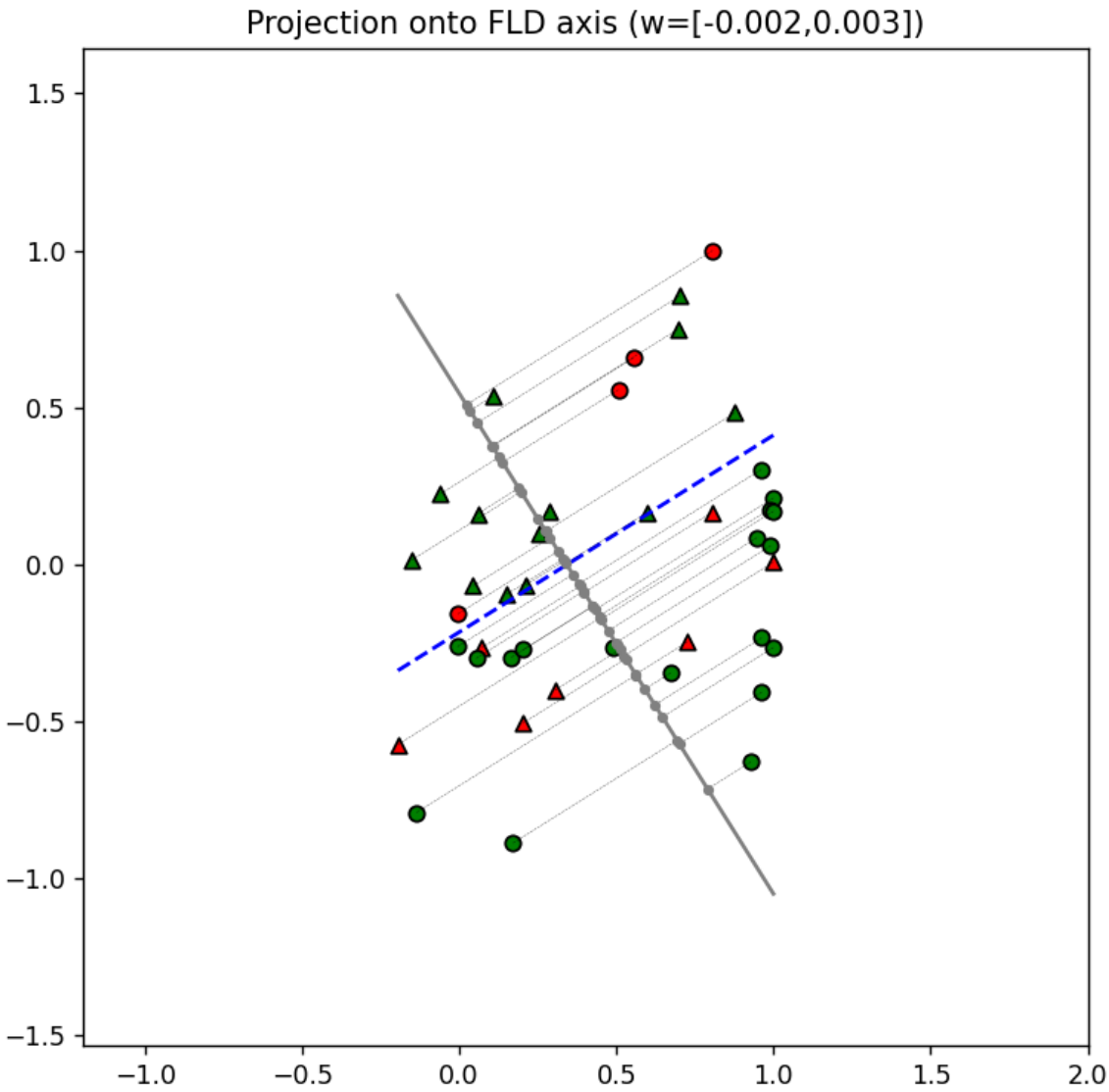
```
2025-10-25 02:05:14.686 | INFO     | __main__:main:238 - FLD: m0=[ 0.35994138 -0.04560139], m
1=[0.32519126 0.04435118] of cols=['27', '30']
2025-10-25 02:05:14.686 | INFO     | __main__:main:239 - FLD:
Sw=
[[41.93041055 15.7202037 ]
 [15.7202037  37.25186904]]
2025-10-25 02:05:14.687 | INFO     | __main__:main:240 - FLD:
Sb=
[[ 0.00120757 -0.00312586]
 [-0.00312586  0.00809147]]
```

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

```
2025-10-25 02:05:14.687 | INFO     | __main__:main:241 - FLD:
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**).

```
2025-10-25 02:05:14.687 | INFO     | __main__:main:242 - FLD: Accuracy=0.7381
```



(10%) Code Check and Verification

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

```
PS D:\ML\hw2> flake8 main.py
PS D:\ML\hw2>
```

```
PS D:\ML\hw2> pytest test_main.py -s
============================================= test session starts =============================================
platform win32 -- Python 3.8.10, pytest-8.3.5, pluggy-1.5.0
rootdir: D:\ML\hw2
collected 2 items

test_main.py (395, 2) (395,)
2025-10-25 02:18:54.371 | INFO     | test_main:test_logistic_regression:35 - accuracy=0.9517
.(395, 2) (395,)
2025-10-25 02:18:54.375 | INFO     | test_main:test_fld:45 - accuracy=0.8759
.

============================================== 2 passed in 11.24s ==============================================
PS D:\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.

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

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

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

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

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

Write your answer here.

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

$$P(x|C_k) = \frac{1}{2\pi^{(D/2)}} \times \frac{1}{|\Sigma|^{(1/2)}} e^{-\frac{1}{2}(x-u_k)^T \cdot \Sigma^{-1}(x-u_k)}$$

$$a = \ln\left(\frac{e^{-\frac{1}{2}(x-u_1)^T\Sigma^{-1}(x-u_1)}}{e^{-\frac{1}{2}(x-u_2)^T\Sigma^{-1}(x-u_2)}}\right) + \ln\frac{P(C_1)}{P(C_2)}$$

$$= -\frac{1}{2}(x-u_1)^T\Sigma^{-1}(x-u_1) + \frac{1}{2}(x-u_2)^T\Sigma^{-1}(x-u_2) + \ln\frac{P(C_1)}{P(C_2)}$$

$$= -\frac{1}{2}(x^T\Sigma^{-1}x - x^T\Sigma^{-1}u_1 - u_1{}^T\Sigma^{-1}x + u_1{}^T\Sigma^{-1}u_1) + \frac{1}{2}(x^T\Sigma^{-1}x - x^T\Sigma^{-1}u_2 - u_2{}^T\Sigma^{-1}x + u_2{}^T\Sigma^{-1}u_2) + \ln\frac{P(C_1)}{P(C_2)}$$

$$= \frac{1}{2}x^T\Sigma^{-1}(u_1 - u_2) + \frac{1}{2}(u_1{}^T - u_2{}^T)\Sigma^{-1}x - \frac{1}{2}u_1{}^T\Sigma^{-1}u_1 + \frac{1}{2}u_2{}^T\Sigma^{-1}u_2 + \ln\frac{P(C_1)}{P(C_2)}$$

$\because x^T\Sigma^{-1}(u_1 - u_2)$ is a value and $(\Sigma^{-1})^T = \Sigma^{-1}$ $\therefore x^T\Sigma^{-1}(u_1 - u_2) = \frac{1}{2}(u_1{}^T - u_2{}^T)\Sigma^{-1}x$

$$= (u_1{}^T - u_2{}^T)\Sigma^{-1}x - \frac{1}{2}u_1{}^T\Sigma^{-1}u_1 + \frac{1}{2}u_2{}^T\Sigma^{-1}u_2 + \ln\frac{P(C_1)}{P(C_2)}$$

$$= (\Sigma^{-1}(u_1 - u_2))^T x - \frac{1}{2}u_1{}^T\Sigma^{-1}u_1 + \frac{1}{2}u_2{}^T\Sigma^{-1}u_2 + \ln\frac{P(C_1)}{P(C_2)}$$

$w^T x$ $\qquad$ w_0

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.”

Write your answer here.

(a) situation: Classifying visually similar insects. why: The differences between classes are very subtle and difficult for humans to distinguish. LR maps the input features to probabilities, providing a smoother and more reliable classification than the Perceptron, which only outputs hard binary labels.

(b) No, it primarily solves a binary classification problem. LR uses a linear combination of input features followed by a logistic (sigmoid) function to map values to the range [0,1]. Although the term "regression" appears in its name, it refers to the linear combination of features, not predicting continuous values.

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.

Write your answer here.

(a)

Features with different scales can cause gradient updates to vary widely, slowing down convergence. Scaling ensures all features contribute proportionally.

Large feature values can cause overflow or underflow in the sigmoid function, leading to unstable training. Scaling mitigates this issue.

(b)

Features with large magnitudes can dominate weight updates, causing oscillations or slow convergence.

Large inputs to the sigmoid function can saturate the output (close to 0 or 1), resulting in vanishing gradients.

Features with larger scales dominate the loss function, potentially causing the model to ignore other important features.