NYCU Introduction to Machine Learning, Homework 2

111550108, 吳佳諭

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-4,  # Y
    num_iterations=1000000,
)
```

```
2024-10-17 22:04:20.255 | INFO     | __main__:main:137 - LR: Weights: [-0.50074993
 0.1088365   0.60821104  0.06065053  0.13522592], Intercep: -1.814744450356557
```

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

```
AUC=0.8500
```

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

```
2024-10-17 22:04:20.255 | INFO     | __main__:main:138 - LR: Accuracy=0.8095
```

(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.

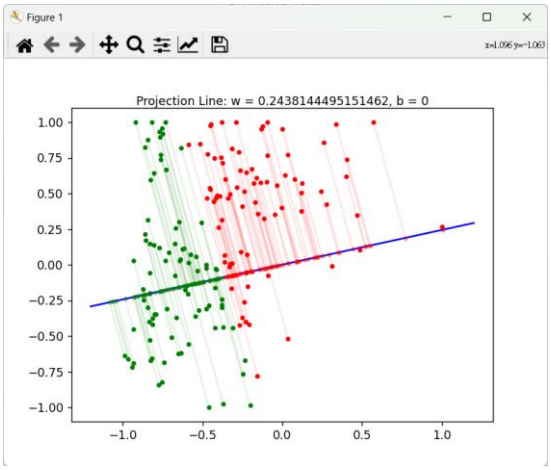
```
FLD: m0=[-0.27747695  0.29565197], m1=[-0.58535466  0.02331584] of cols=['10', '20']
2024-10-18 22:27:41.482 | INFO     | __main__:main:209 - FLD:
Sw=
[[17.17974856  5.44299487]
 [ 5.44299487 44.81848741]]
2024-10-18 22:27:41.483 | INFO     | __main__:main:210 - FLD:
Sb=
[[0.09478869 0.08384622]
 [0.08384622 0.07416696]]
```

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

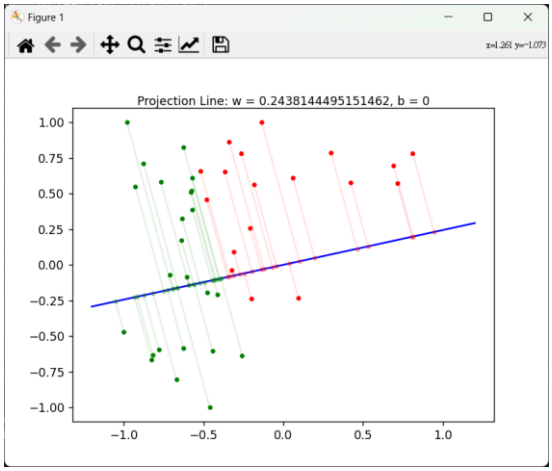
```
2024-10-18 22:27:41.484 | INFO     | __main__:main:211 - FLD:
W=
[0.0166359  0.00405607]
```

6. (15%) 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. (Also, plot for training data). Show the accuracy score on the testing set.

```
2024-10-18 22:27:41.485 | INFO     | __main__:main:212 - FLD: Accuracy=0.7619
```



Plot with x_train



Plot with x_test

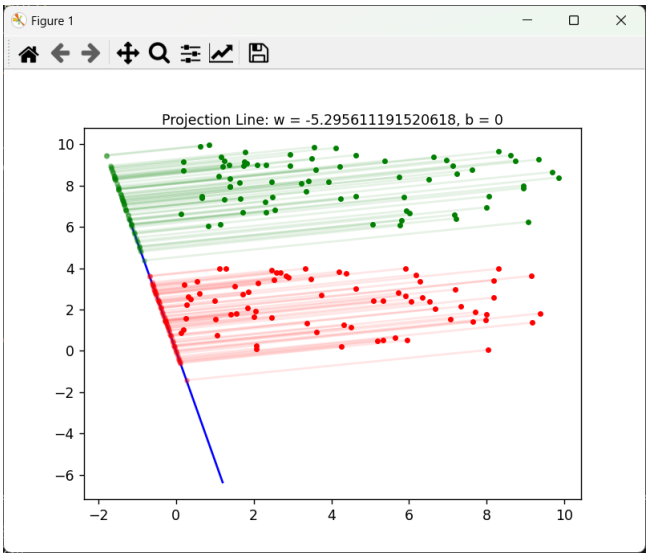
(10%) Code Check and Verification

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

```
PS D:\佳佳\交大\機器學習\HW2> flake8 main.py
PS D:\佳佳\交大\機器學習\HW2> pytest ./test_main.py -s
============================= test session starts =============================
platform win32 -- Python 3.11.7, pytest-7.4.0, pluggy-1.0.0
rootdir: D:\佳佳\交大\機器學習\HW2
plugins: anyio-4.2.0
collected 2 items

test_main.py (395, 2) (395,)
2024-10-24 09:08:56.788 | INFO     | test_main:test_logistic_regression:35 - accuracy=0.9517
.(395, 2) (395,)
2024-10-24 09:08:56.799 | INFO     | test_main:test_fld:45 - accuracy=0.8759
.

============================== 2 passed in 40.30s ==============================
```



Part. 2, Questions (40%):

1. **(10%)**
   - **Is logistic 'regression' used for regression problems?**
   - **If not, what task is it primarily used? (without any additional techniques and modification); If yes, how can it be implemented?**
   - **Why are we using the logistic function in such a task? (list two reasons)**
   - **If there are multi-class, what should we use to substitute it?**

No, logistic regression is for classification task.

We use the logistic function because it can be directly interpreted as the probability of the class (the output is always 0~1). And it is differentiable, which makes it simpler to implement. Secondly, it is better able to catch the subtleties data such as some nonlinear correlation, so it is more precise. Last but not least, logistic function is less likely to overfit. Therefore, it can give a better outcome.

In a multi-class classification, we should substitute it with softmax $\frac{\exp(a_k)}{\sum_j \exp(a_j)}$, where $a_k = \ln [p(x|C_k)p(C_k)]$

2. **(15%) When a trained classification model shows exceptionally high precision but unusually low recall and F1-score, what potential issues might arise? How can these issues be resolved? List at least three solutions.**

It implies that the model is overfitting. When it is training, it can classify the data very well, making us think we have a great model. However, when it sees new data, it cannot classify the data correctly, leading to a bad performance. Or the threshold is too high, so it will have a high precision and low recall. Or the dataset has more data of class0 than class1, then the model will focus on class0 and ignore class1.

To solve the problem, we can modify the threshold or learning rate, or use some regularization techniques, so the model will not overfit the training data. Also, we can adjust the dataset. If there are more data in class0, we can add some data to class1. Or give class1 a weight, so the model will give more importance to class1 and learn class1 better.

3. (15%) In this homework, we use Cross-Entropy as the loss function for Logistic Regression. Can we use Mean Square Error (MSE) instead? Why or why not? Please explain in detail.

No, we can't use MSE in classification. First, MSE assumes the data has a Gaussian distribution, but in classification, data is in two classes, which is a Bernoulli distribution, so it is not suitable to use MSE. Second, MSE function in classification is non-convex. Gradient descent is not guaranteed to find the minimal point. Third, CE can give more penalty than MSE if the model makes a wrong classification. If we make a wrong classification, the MSE will be $(1-0)^2 = 1$, but cross entropy will be $-(0 * ln0 + 0 * ln1) = -\infty$, which can make the model learn better. Overall, cross entropy can performance better than MSE, and MSE may not be able to get the correct result, so we use cross entropy in logistic regression.