NYCU Introduction to Machine Learning, Homework 2

[111550149], [林悅揚]

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=0.1,  # You can modify the parameters as you want
    num_iterations=1000,  # You can modify the parameters as you want
)
```

```
2024-10-25 13:56:19.821 | INFO     | __main__:main:124 - LR: Weights: [-0.7653145   0.10389985  1.01565643 -0.10632914  0.07557573], Intercep: -2.5901763492591323
```

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

```
2024-10-25 13:56:19.823 | INFO     | __main__:main:125 - LR: Accuracy=0.8333, AUC=0.8614
```

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

```
2024-10-25 13:56:19.823 | INFO     | __main__:main:125 - LR: Accuracy=0.8333, AUC=0.8614
```

(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.

```
2024-10-25 21:01:46.261 | INFO     | __main__:main:136 - FLD: m0=[-0.27747695  0.29565197], m1=[-0.58535466  0.02331584] of cols=['10', '20']
2024-10-25 21:01:46.261 | INFO     | __main__:main:137 - FLD:
Sw=
[[17.17974856  5.44299487]
 [ 5.44299487 44.81848741]]
2024-10-25 21:01:46.261 | INFO     | __main__:main:138 - FLD:
Sb=
[[0.09478869 0.08384622]
 [0.08384622 0.07416696]]
```

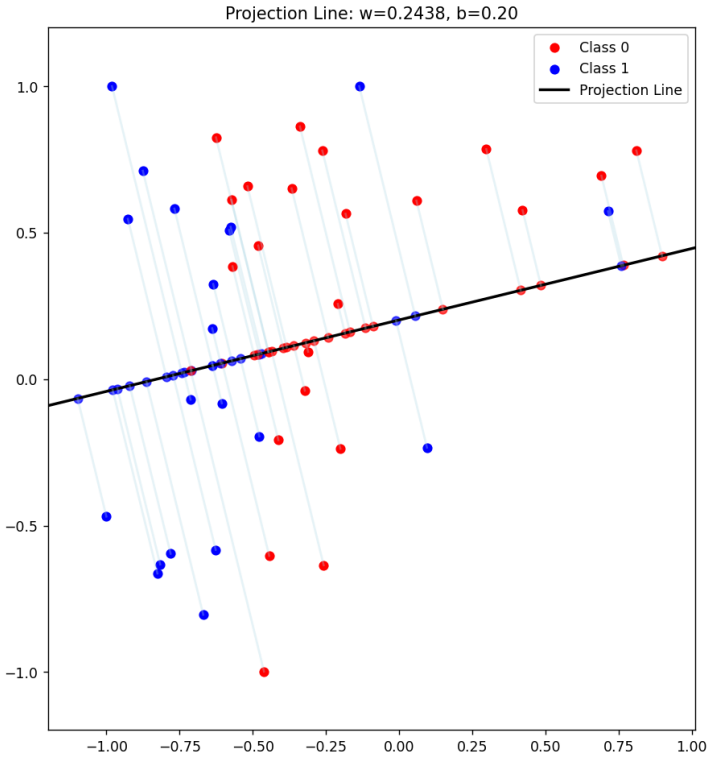
5. (5%) Show the Fisher's linear discriminant *w* of the training set.

```
2024-10-25 21:01:46.261 | INFO     | __main__:main:139 - FLD:
w=
[-0.97154001 -0.23687549]
```

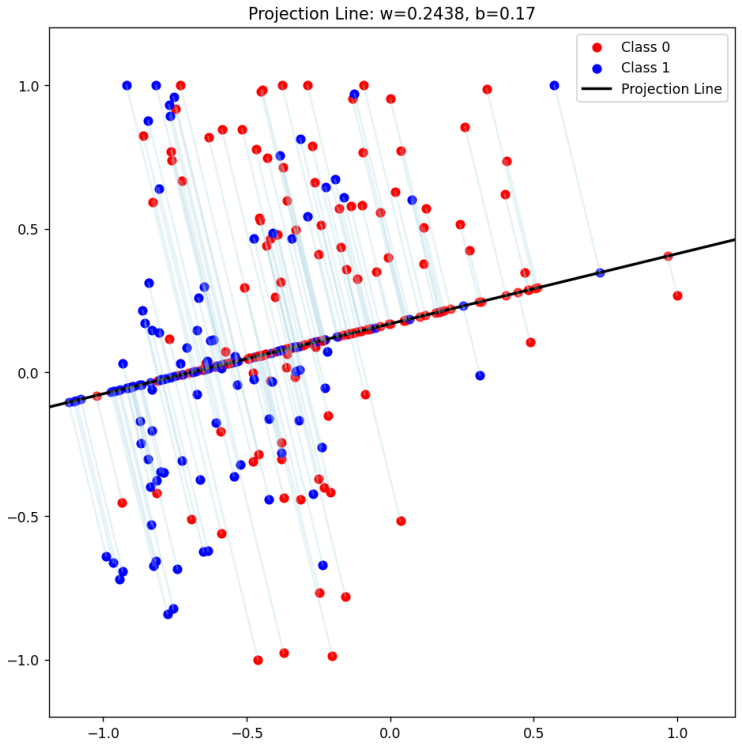
6. (15%) Obtain predictions for the testing set by measuring the distance between the projected value of the testing data and the projected means of the training data for the two classes. (Also, plot for training data). Show the accuracy score on the testing set.

```
2024-10-25 21:01:46.261 | INFO     | __main__:main:140 - FLD: Accuracy=0.7619
```

Testing data:



Training data:



(10%) Code Check and Verification

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

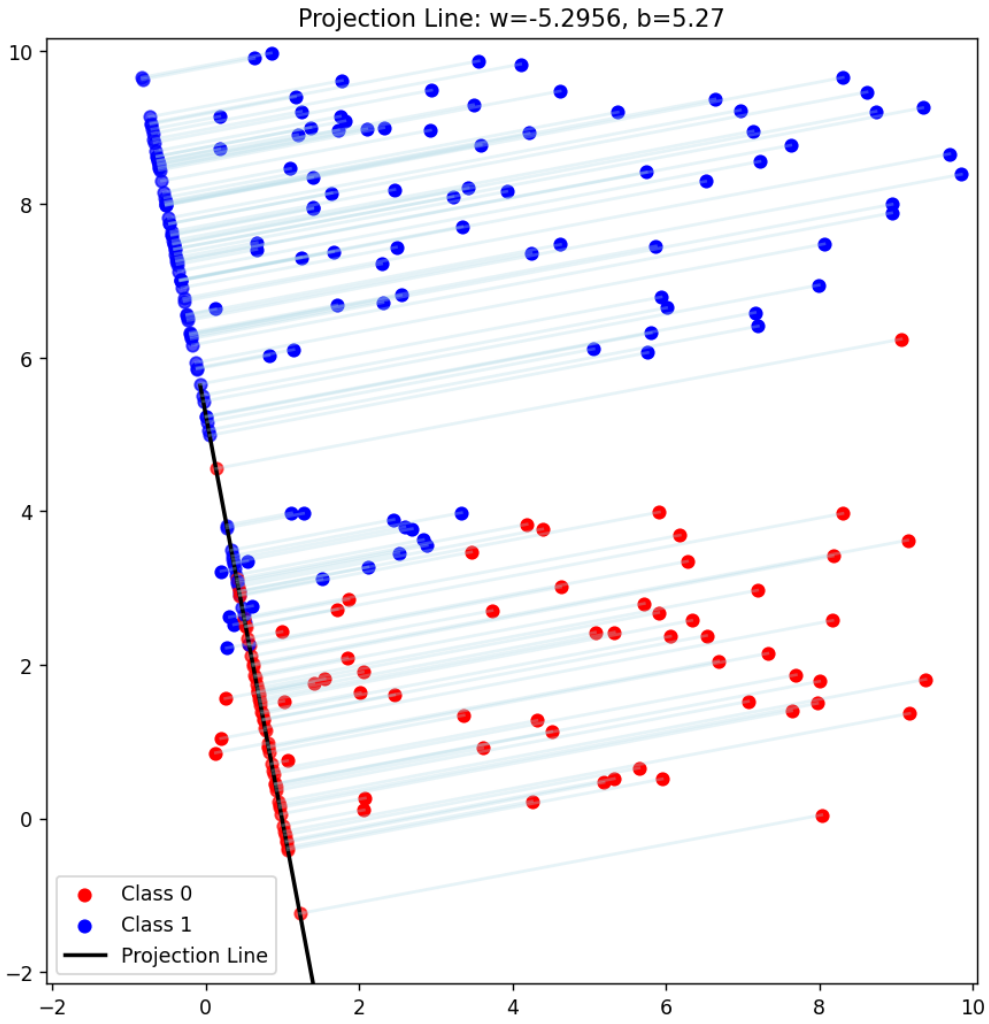
```
PS C:\Users\USER\Desktop\hw\ml\hw2\release (1)> flake8 .\main.py
PS C:\Users\USER\Desktop\hw\ml\hw2\release (1)>
```

```
PS C:\Users\USER\Desktop\hw\ml\hw2\release (1)> pytest .\test_main.py -s
============================= test session starts =============================
platform win32 -- Python 3.9.19, pytest-7.4.4, pluggy-1.0.0
rootdir: C:\Users\USER\Desktop\hw\ml\hw2\release (1)
collected 2 items

test_main.py (395, 2) (395,)
2024-10-25 21:10:44.101 | INFO     | test_main:test_logistic_regression:35 - accuracy=0.9586
.(395, 2) (395,)
2024-10-25 21:10:44.120 | INFO     | test_main:test_fld:45 - accuracy=0.8759
.

============================== 2 passed in 3.18s ==============================
```

x and y show np.min-2 rather than np.min-0.2



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?

   1. Logistic regression is mainly used for classification rather than regression tasks.
   2. Logicstic regression is primarily used for binary classification tasks.
   3. ◎The logic function like sigmoid restrict the output value between 0 and 1.It is suitable for probabilities of binary classification.
      ◎It maps the linear combination of input to a non-linear space. Enabling the model to better distinguish between the classes, especially in binary classification tasks.
   4.Multinomial logistic regression extends binary logistic regression to manage multi-class problems directly, eliminating the need for one-vs-rest or one-vs-one strategies. This approach uses a single model to predict multiple output classes, estimating parameters for eac h class at the same time. It generalizes binary logistic regression by adding specific parameters for each class.

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

   The model may focus excessively on identifying the minority class correctly. If one class i s significantly underrepresented, the model might lack sensitivity to this class. Additionall y, if the model has learned only certain defining features, it may lead to overfitting on spec ific features, resulting in low recall and F1-score.

   1.By adjusting class weights, the model can give more emphasis to the minority class duri ng predictions, increasing recall and balanced metrics.

   2.Over-sampling the minority class or under-sampling the majority class can balance the d ataset.

   3.Increasing the model complexity to improve performance and better identify challenging positives.

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

   It is not recommended for several reason:

   1.Logistic regression is fundamentally a classification model that assumes a binomial or multinomi al distribution of the output. Cross-Entropy loss aligns with this distribution assumption, making it a natural choice. In contrast, MSE assumes the errors are normally distributed (Gaussian), which is

suitable for regression tasks, not classification. Thus, using MSE for logistic regression is theoretic ally inconsistent with the assumptions of the model.

2. Cross-Entropy, in combination with the logistic (sigmoid) function, is designed to predic t probabilities. MSE does not naturally fit with probability predictions in classification task s. Using MSE can cause the output probabilities to deviate significantly from actual probab ilities, making threshold-based classification less reliable.

3. Cross-Entropy loss in logistic regression leads to a convex optimization problem. This c onvexity ensures that gradient descent will more reliably reach the global minimum, resulti ng in stable and optimal training. However, using MSE may lead to a non-convex loss lan dscape in logistic regression. This non-convexity increases the risk of the optimization pro cess getting stuck in local minima, which can result in poor performance.