NYCU Introduction to Machine Learning, Homework 2

110550130, 劉秉驊

Part. 1, Coding (50%):

In this coding assignment, you are requested to implement Logistic Regression and Fisher's Linear Discriminant by using only Numpy. After that, train your model on the provided dataset and evaluate the performance on the testing data.

(15%) Logistic Regression

Requirements:

- Use Gradient Descent to update your model
- Use CE (Cross-Entropy) as your loss function.

Criteria:

1. (0%) Show the hyperparameters (learning rate and iteration) that you used.

```
LR = LogisticRegression(learning_rate=0.00015, iteration=100000)
```

2. (5%) Show the weights and intercept of your model.

```
Part 1: Logistic Regression
Weights: [-0.05380495 -0.76703905  0.90886422 -0.04970127  0.0283856  -0.55288738], Intercept: -0.06754147593603659
```

3. (10%) Show the accuracy score of your model on the testing set. The accuracy score should be greater than 0.75.

Last line.

```
Part 1: Logistic Regression
Weights: [-0.05380495 -0.76703905  0.90886422 -0.04970127  0.0283856  -0.55288738], Intercept: -0.06754147593603659
Accuracy: 0.7540983606557377
```

(35%) Fisher's Linear Discriminant (FLD)

Requirements:

- Implement FLD to reduce the dimension of the data from 2-dimensional to 1-dimensional.

Criteria:

4. (0%) Show the mean vectors m_i (i=0, 1) of each class of the training set.

```
Part 2: Fisher's Linear Discriminant
Class Mean 0: [ 56.75925926 137.7962963 ], Class Mean 1: [ 52.63432836 158.97761194]
```

5. (5%) Show the within-class scatter matrix S_W of the training set.

Last matrix.

```
Part 2: Fisher's Linear Discriminant
Class Mean 0: [ 56.75925926 137.7962963 ], Class Mean 1: [ 52.63432836 158.97761194]
With-in class scatter matrix:
[[ 19184.82283029 -16006.39331122]
 [-16006.39331122 106946.45135434]]
```

6. (5%) Show the between-class scatter matrix S_B of the training set.

Last matrix.

```
Part 2: Fisher's Linear Discriminant
Class Mean 0: [ 56.75925926 137.7962963 ], Class Mean 1: [ 52.63432836 158.97761194]
With-in class scatter matrix:
[[ 19184.82283029 -16006.39331122]
 [-16006.39331122 106946.45135434]]
Between class scatter matrix:
[[ 17.01505494 -87.37146342]
 [-87.37146342 448.64813241]]
```

7. (5%) Show the Fisher's linear discriminant *w* of the training set.

Last matrix.

```
Part 2: Fisher's Linear Discriminant
Class Mean 0: [ 56.75925926 137.7962963 ], Class Mean 1: [ 52.63432836 158.97761194]
With-in class scatter matrix:
[[ 19184.82283029 -16006.39331122]
 [-16006.39331122 106946.45135434]]
Between class scatter matrix:
[[ 17.01505494 -87.37146342]
 [-87.37146342 448.64813241]]
w:
[-5.68686969e-05  1.89543951e-04]
```

8. (10%) 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. Show the accuracy score on the testing set. The accuracy score should be greater than 0.65.

Last line.

```
Part 2: Fisher's Linear Discriminant
Class Mean 0: [ 56.75925926 137.7962963 ], Class Mean 1: [ 52.63432836 158.97761194]
With-in class scatter matrix:
[[ 19184.82283029 -16006.39331122]
 [-16006.39331122 106946.45135434]]
Between class scatter matrix:
[[ 17.01505494 -87.37146342]
 [-87.37146342 448.64813241]]
w:
[-5.68686969e-05  1.89543951e-04]
Accuracy of FLD: 0.6557377049180327
```

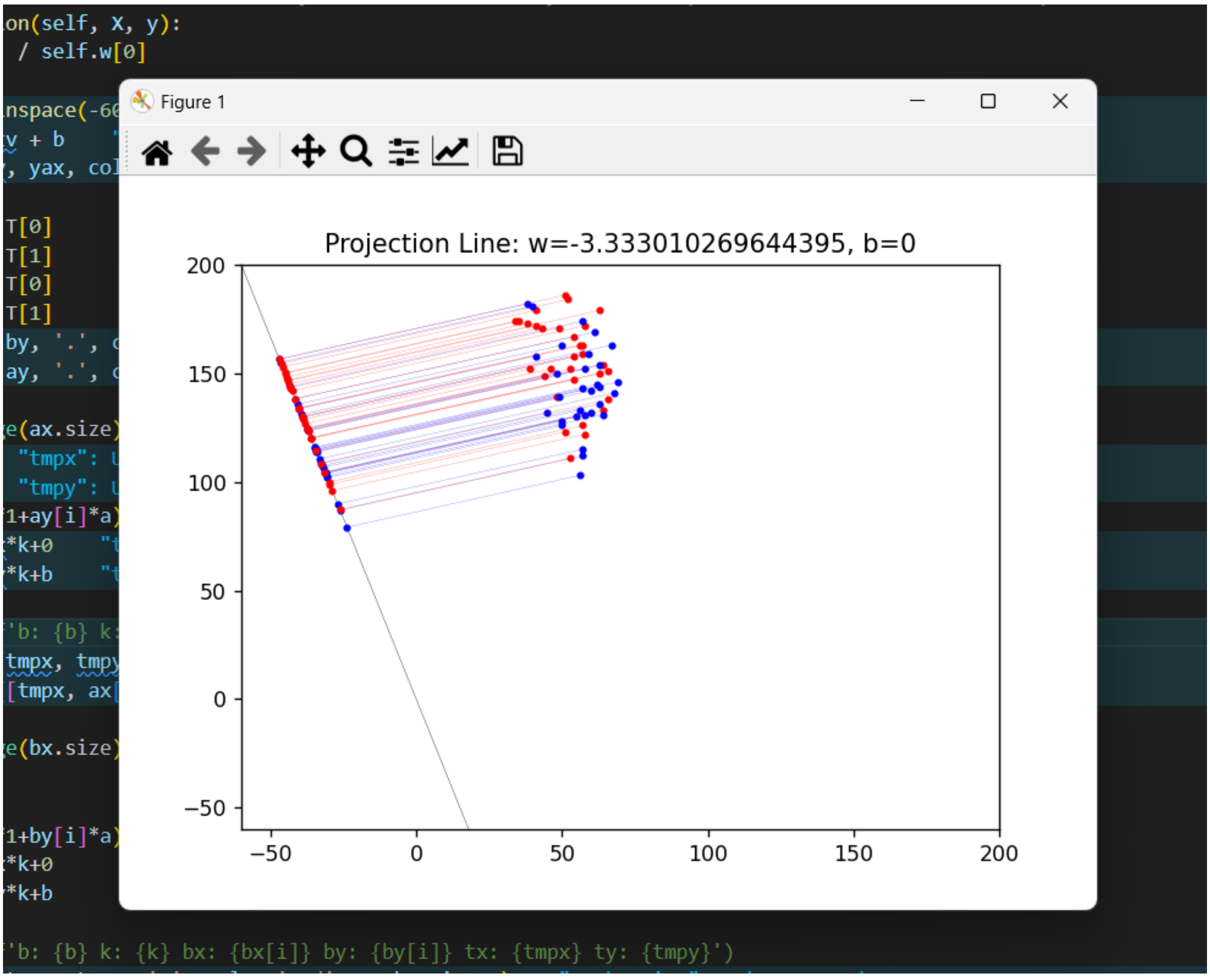
(next page)

9. (10%) Plot the projection line (x-axis: age, y-axis: thalach).

   1) Plot the projection line trained on the training set and show the slope and intercept on the title (you can choose any value of intercept for better visualization).

   2) Obtain the prediction of the testing set, plot and colorize them based on the prediction.

3) Project all testing data points on your projection line. Your result should look like the below image.



Part. 2, Questions (50%):

1. (5%) What's the difference between the sigmoid function and the softmax function? In what scenarios will the two functions be used? Please at least provide one difference for the first question and answer the second question respectively.

The main difference is that sigmoid which usually has only 2 results for true and false is used in binary classification, and, on the other hand, softmax which has a result for each class is used in multiple classification.
For the scenarios, sigmoid can be used in sentiment analysis with logistic regression in which we classify the input to the bigger or smaller class. And softmax can be used in language recognition in which we need to know how possible the input might be in a specified class like language..

2. (10%) In this homework, we use the cross-entropy function as the loss function for Logistic Regression. Why can't we use Mean Square Error (MSE) instead? Please explain in detail.

First, logistic regression uses probability to train data that is confined to distribution of 0 to 1, which cross-entropy function(CEF) can afford. But MSE is suitable to calculate the difference between real and predicted values.
Second, MSE is easy to get affected by error, but CEF is not. If using MSE, we might obtain unexpected outliers.
Third, MSE is of non-convexity that results in convergence with logistic regression, but CEF which is of convexity won't be.

(next page)

3. (15%) In a multi-class classification problem, assume you have already trained a classifier using a logistic regression model, which the outputs are P1, P2, ... Pc, how do you evaluate the overall performance of this classifier with respect to its ability to predict the correct class?

    3.1. (5%) What are the metrics that are commonly used to evaluate the performance of the classifier? Please at least list three of them.

Accuracy which consists of correctly predicted cases divided by all cases.
Precision which consists of true positive observation divided by all positive observation.
Recall which consists of true positive observation divided by true positive plus false negative observation.

    3.2. (5%) Based on the previous question, how do you determine the predicted class of each sample?

Every class has a predicted probability, and we choose the predicted class from the class with the highest probability.

    3.3. (5%) In a class imbalance dataset (say 90% of class-1, 9% of class-2, and 1% of class-3), is there any problem with using the metrics you mentioned above and how to evaluate the model prediction performance in a fair manner?

Except for precision and recall, accuracy might rise unexpectedly. Class 1 is dominant. We can take the tactic as follows: first, set weights for each metric that the weight of accuracy takes lower to reduce the impact of imbalance accuracy. Second, plot ROC-AUC to observe performance of each class with different thresholds that helps analyze independently.

4. (20%) Calculate the results of the partial derivatives for the following equations. (The first one is binary cross-entropy loss, and the second one is mean square error loss followed by a sigmoid function. σ is the sigmoid function.)

4.1. (10%)

$$\frac{\partial}{\partial x}\left(-t * \ln\big(\sigma(x)\big) - (1-t) * \ln\big(1-\sigma(x)\big)\right)$$

$$\sigma(x) = (1+e^{-x})^{-1}$$

$$\frac{d}{dx}\sigma(x) = \frac{e^{-x}}{(1+e^{-x})^2} = e^{-x}(\sigma(x))^2$$

$$\frac{d}{dx}\left(-t\ln(\sigma(x)) - (1-t)\ln(1-\sigma(x))\right)$$

$$= -t\frac{1}{\sigma(x)}\frac{d}{dx}\sigma(x)\frac{d}{dx}x - (1-t)\frac{1}{1-\sigma(x)}\frac{d}{dx}(1-\sigma(x))$$

$$= \frac{-t}{\sigma(x)}\frac{d}{dx}\sigma(x) - \frac{(1-t)}{1-\sigma(x)}\left(-\frac{d}{dx}\sigma(x)\frac{d}{dx}x\right)$$

$$= \left[\frac{d}{dx}\sigma(x)\right]\left[\frac{-t}{\sigma(x)} + \frac{1-t}{1-\sigma(x)}\right]$$

$$= e^{-x}\sigma^2(x)\left[\frac{-t}{\sigma(x)} + \frac{1-t}{1-\sigma(x)}\right] \#$$

4.2. (10%)

$$\frac{\partial}{\partial x}\left(\left(t-\sigma(x)\right)^2\right)$$

$$\frac{d}{dx}(t-\sigma(x))^2$$

$$= 2[t-\sigma(x)]\frac{d}{dx}(t-\sigma(x))$$

$$= -2[t-\sigma(x)]\left[\frac{d}{dx}\sigma(x)\right]$$

$$= -2e^{-x}\sigma^2(x)[t-\sigma(x)] \#$$