NYCU Introduction to Machine Learning, Homework 2

110705013, 沈昱宏

Part. 1, Coding (50%):

(15%) Logistic Regression

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

```
LR = LogisticRegression(learning_rate=0.001, iteration=130000)
```

```
if(i==125000):
    self.learning_rate /= 2
```

learning_rate : 0.001 for the first 125000 and 0.0005 for subsequent 5000.

2. (5%)(10%) Show the weights and intercept of your model. & Show the accuracy score of your model on the testing set.

```
Part 1: Logistic Regression
Weights: [-0.07198652 -3.04935559  1.91498458 -0.39996029  0.05324534 -1.30234743], Intercept: -0.30176200967689804
Accuracy: 0.7704918032786885
```

3. (in 2.)

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

(full screenshot)

```
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:
[[ 156.60777894 -124.17509578]
 [-124.17509578  910.55595365]]
Between class scatter matrix:
[[ 17.01505494 -87.37146342]
 [-87.37146342 448.64813241]]
w:
[-0.00885188  0.0220548 ]
Accuracy of FLD: 0.6557377049180327
```

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

```
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. W

```
With-in class scatter matrix:
[[ 156.60777894 -124.17509578]
 [-124.17509578  910.55595365]]
```

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

```
Between class scatter matrix:
[[ 17.01505494 -87.37146342]
 [-87.37146342 448.64813241]]
```

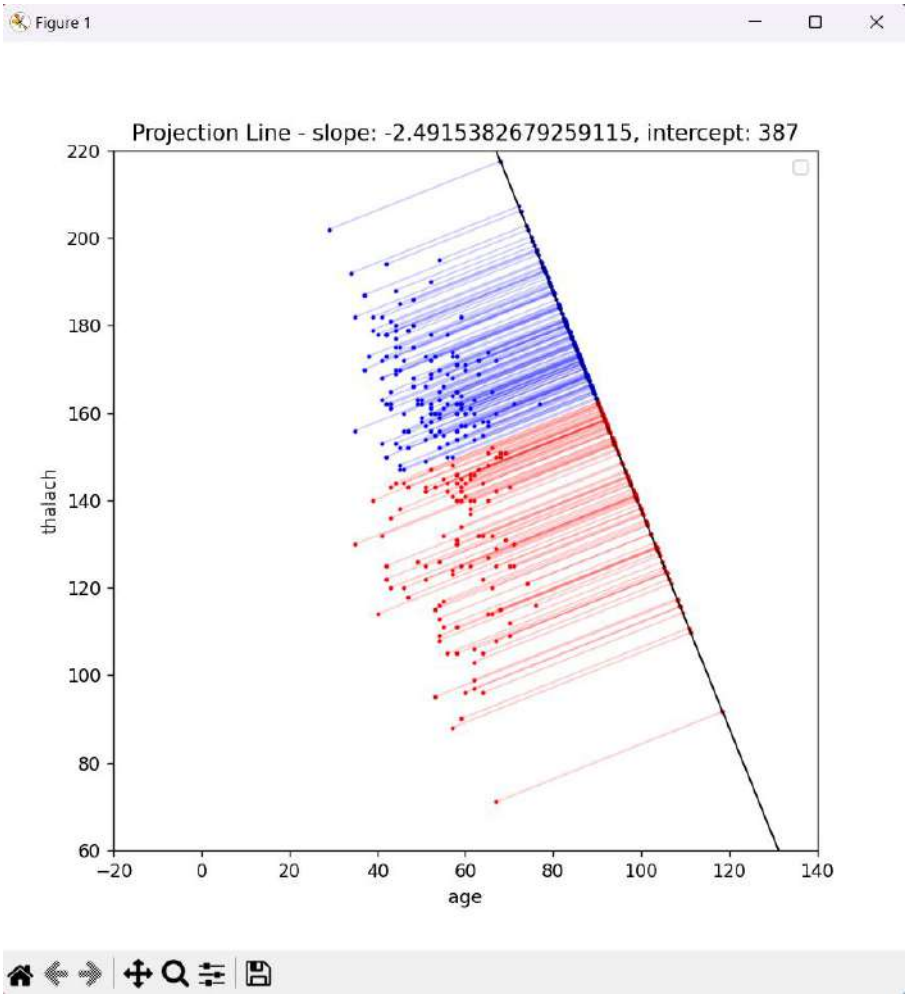
7. (5%) Show the Fisher's linear discriminant w of the training set.

```
w:
[-0.00885188  0.0220548 ]
```

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.

```
Accuracy of FLD: 0.6557377049180327
```

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



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 sigmoid function takes a value as input and transform the value in (0,1) as output.

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

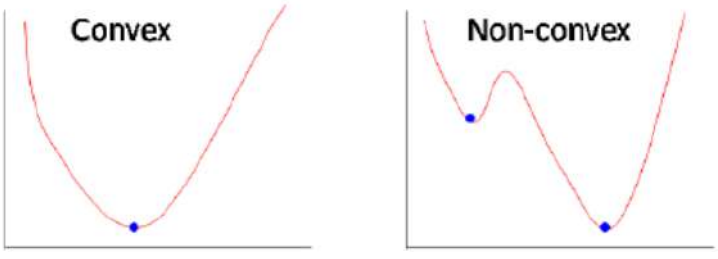
   The softmax function takes a vector as input, and it outputs a probability distribution.

   $$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad \text{for } i = 1, \ldots, K \text{ and } \mathbf{z} = (z_1, \ldots, z_K) \in \mathbb{R}^K.$$

   The sigmoid function is used for classification problem where there are only two classes to classify.
   The softmax function is used for multi-class classification problem.

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.

   1. MSE assumes that data our sampled from a Gaussian distribution. In our homework, we assume that the data comes from two categories, target = 0 or 1, which is not a Gaussian distribution.
   2. MSE lead to non-convex function in classification. A function is convex iff the function's second derivative is always >= 0. If a function is non-convex, it might converge to local minimum or saddle points instead of global minimum, which is the other reason why we do not use MSE as the loss function.

   To prove the non-convixity of a function, we can find the second derivative of the function and check if there are points have value less than zero. (here is the prove provided by MSE Loss in Classification)

The second derivative is shown in the following graph. The y_hat denotes the predicted result.

$$\frac{\partial^2 f}{\partial \theta^2} = \frac{\partial}{\partial \theta}\left(\frac{\partial f}{\partial \theta}\right)$$
$$= \frac{\partial}{\partial \theta}\left(\frac{\partial f}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial \theta}\right)$$
$$= -2(y - 2y\hat{y} - 2\hat{y} + 3\hat{y}^2)xx\hat{y}(1-\hat{y})$$

Always positive

And there are conditions that the second directive is smaller than zero, hence the non-convexity of the objective function using MSE is proven.

When $y = 0$

$$H(\hat{y}) = -2(3\hat{y}^2 - 2\hat{y})$$
$$= -2\left[3\hat{y}\left(\hat{y} - \frac{2}{3}\right)\right]$$

Negative when: $\hat{y} \in \left[\frac{2}{3}, 1\right]$

When $y = 1$

$$H(\hat{y}) = -2(3\hat{y}^2 - 4\hat{y} + 1)$$
$$= -6\left(\hat{y} - \frac{1}{3}\right)(\hat{y} - 1)$$

Negative when: $\hat{y} \in \left[0, \frac{1}{3}\right]$

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

3.1 (1) Accuracy : correct prediction / total prediction
(2) Precision : true poisitve / (true positive + false positive), focus on how much your positive prediction is correct.
(3) Recall : true positive / (true positive + false negative), is the ratio of positive in correctly predicted samples.
(4) F1 score : 2 * (precision * recall) / (precision + recall)
For multi-class, you can calculate the (2)(3)(4) for each class and average them.

3.2

We typically choose the class with highest P as prediction, which is $\text{argmax}_i(P_i)$. For prediction class i, if the truth is of the sample is i, it is considered true positive when calculating for i, and it is true negative when calculating other classes j!=i. For prediction class i, if the truth of sample is j, it is considered false positive when calculating for i, false negative when calculating for j, and true negative when calculating other classes k!=(i or j).

3.3

If we have a dataset with 90% of class-1, 9% of class-2, and 1% of class-3, a classifier that always predicts class-1 will have a high accuracy(90%) but will not be useful in practice. In such cases, we can use the average of recall obtained on each class. If the prediction was 100% class-1, the recall of class-2 and class-3 is hoing to be 0, and the recall of class-1 is 100%, thus the average recall is 33%, indicating the model is not good.

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%)

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

4.2. (10%)

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

4.1

$= -t\frac{\sigma'(x)}{\sigma(x)} - (1-t)\frac{-\sigma'(x)}{1-\sigma(x)}$

$= -t\frac{\sigma'(x)}{\sigma(x)} + (1-t)\sigma(x)$ ①

$= -t\,\sigma(x)e^{-x} + (1-t)\sigma(x)$ ②

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

$= -t + \sigma(x)$ #

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

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

$\frac{\sigma'(x)}{1-\sigma(x)} = \frac{e^{-x}\times(1+e^{-x})}{(1+e^{-x})^2 \times e^{-x}} = \sigma(x)$ ... ①

$\frac{\sigma'(x)}{\sigma(x)} = \frac{e^{-x}\times(1+e^{-x})}{(1+e^{-x})^2} = \frac{e^{-x}}{1+e^{-x}} = \sigma(x)e^{-x}$ ... ②

4.2

$= 2(t-\sigma(x)) \times -\sigma'(x)$

$= -2(t-\sigma(x)) \cdot (1-\sigma(x)) \times \sigma(x)$ #