NYCU Introduction to Machine Learning, Homework 2

110652021, 龔大承

The screenshot and the figures we provided below are just examples. **The results below are not guaranteed to be correct.** Please make sure your answers are clear and readable, or no points will be given. Please also remember to convert it to a pdf file before submission. You should use English to answer the questions. After reading this paragraph, you can delete this paragraph.

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

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

```
LR = LogisticRegression(learning_rate=0.0001, iteration=500000)
```

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

```
Weights: [-0.05520355 -1.51659289  1.02393276 -0.18381133  0.03402888 -0.66696971], Intercept: -0.12931391957660726
```

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

```
Accuracy: 0.7540983606557377
```

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

4. (0%) Show the mean vectors mi (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 SW of the training set.

```
With-in class scatter matrix:
[[ 19184.82283029 -16006.39331122]
 [-16006.39331122 106946.45135434]]
```

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

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

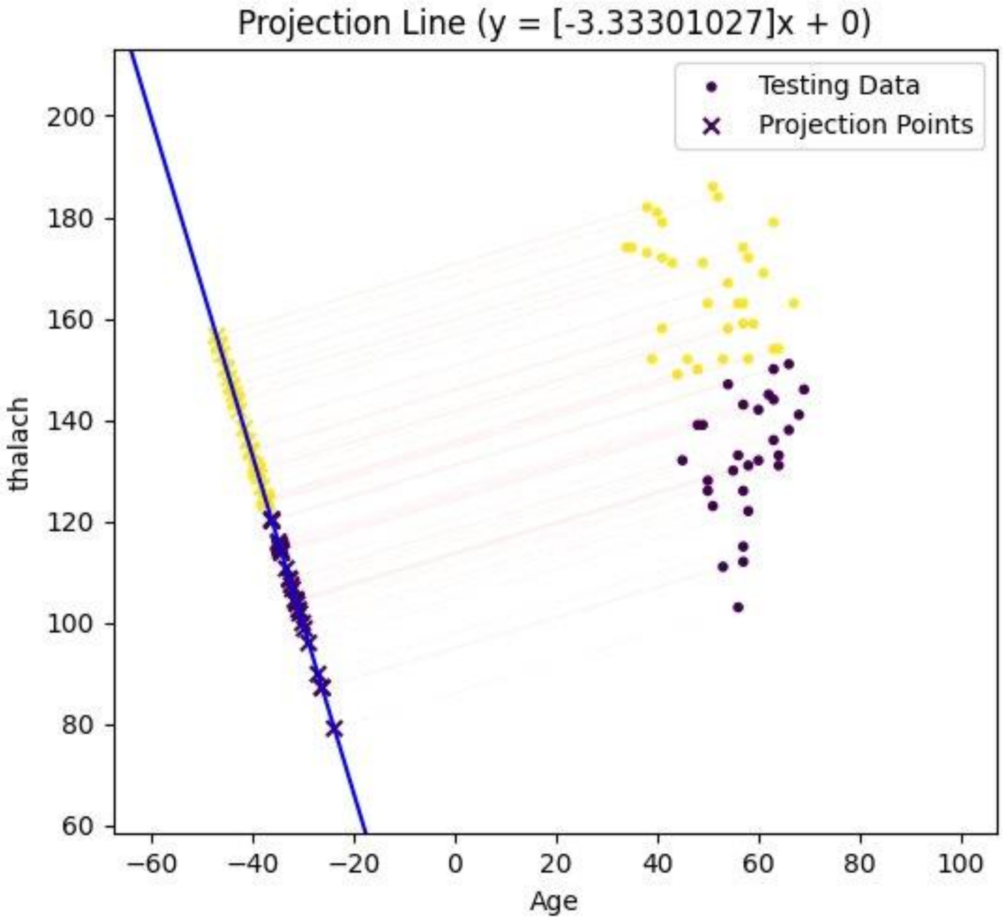
7. (5%) Show the Fisher's linear discriminant w of the training set.

```
w:
[-0.28737344  0.95781862]
```

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.

```
Accuracy of FLD: 0.6557377049180327
```

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.

Difference:

Sigmoid Function: Used for binary classification (2 classes), squashes values to a range of 0 to 1.

Softmax Function: Used for multi-class classification (more than 2 classes), normalizes outputs to probabilities summing to 1.

Scenarios:

Sigmoid: Applied in binary tasks like spam detection or sentiment analysis.

Softmax: Employed in multi-class problems like image classification or natural language processing tasks.

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.

Using Mean Square Error (MSE) as the loss function in logistic regression is not appropriate for several reasons:

(1). Output Range Mismatch:

- MSE: Expects continuous output values.
- Logistic Regression: Outputs probabilities between 0 and 1.

(2). Non-Convex Optimization:

- MSE: Leads to non-convex optimization problems.
- Logistic Regression: Requires convex optimization for reliable convergence.

(3). Sensitivity to Outliers:

- MSE: Sensitive to outliers, impacting model performance.
- Logistic Regression: Should be robust to outliers for better generalization.

(4). Gradient Descent Challenges:

- MSE: Gradient behavior can cause slow learning.
- Logistic Regression: Needs informative gradients for faster, stable convergence.

In summary, logistic regression benefits from using the cross-entropy loss due to its suitability for probability outputs, convex optimization, robustness to outliers, and compatibility with efficient gradient descent.

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?

1. (5%) What are the metrics that are commonly used to evaluate the performance of the classifier? Please at least list three of them.

   Three commonly used metrics for evaluating the performance of a multi-class classifier are:

   Accuracy: It measures the proportion of correctly classified samples out of the total samples.

   Precision: Precision is the ratio of correctly predicted positive observations to the total predicted positives. It is especially relevant in situations where false positives are costly.

   Recall (Sensitivity): Recall is the ratio of correctly predicted positive observations to the all observations in actual class. It is crucial when false negatives are costly.
2. (5%) Based on the previous question, how do you determine the predicted class of each sample?

   The predicted class for each sample is determined by selecting the class with the highest predicted probability. In other words, if P1 has the highest probability among P1,P2,...,Pc, then the predicted class is class 1. This is typically done after applying a softmax operation to convert raw model outputs into probabilities.
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?

   In a class-imbalanced dataset, using metrics like accuracy can be misleading because the model might perform well on the majority class but poorly on minority classes. To address this issue, consider using:

Precision-Recall Curves: Plotting precision against recall for different probability thresholds can provide insights, especially when dealing with imbalanced datasets.

F1 Score: The harmonic mean of precision and recall, it balances both metrics and is useful in imbalanced scenarios.

Confusion Matrix: Examining the confusion matrix can reveal how well the model is performing for each class, allowing for a more detailed assessment.

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.)

1.(10%)

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

2.(10%)

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

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

$\frac{\partial}{\partial x}\left(\frac{1}{1+e^{-x}}\right)$

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

$= \sigma(x)(1-\sigma(x))$

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

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

$= -t\frac{\cancel{\sigma(x)}(1-\sigma(x))}{\cancel{\sigma(x)}} + (1-t)\frac{\sigma(x)\cancel{(1-\sigma(x))}}{\cancel{1-\sigma(x)}}$

$= -t(1-\sigma(x)) + (1-t)\sigma(x)$

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

$= \sigma(x) - t$

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

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

$= -2(t-\sigma(x))\sigma(x)(1-\sigma(x))$