NYCU Pattern Recognition, Homework 2

**Deadline: Apr. 05, 23:59**

**Part. 1, Coding (70%)**:

In this coding assignment, you are requested to implement 1) logistic Regression and 2) Fisher’s Linear Discriminant by using only Numpy, then train your model on the provided dataset and finally evaluate the performance on testing data. Please train your logistic regression model using Gradient Descent, not the closed-form solution.

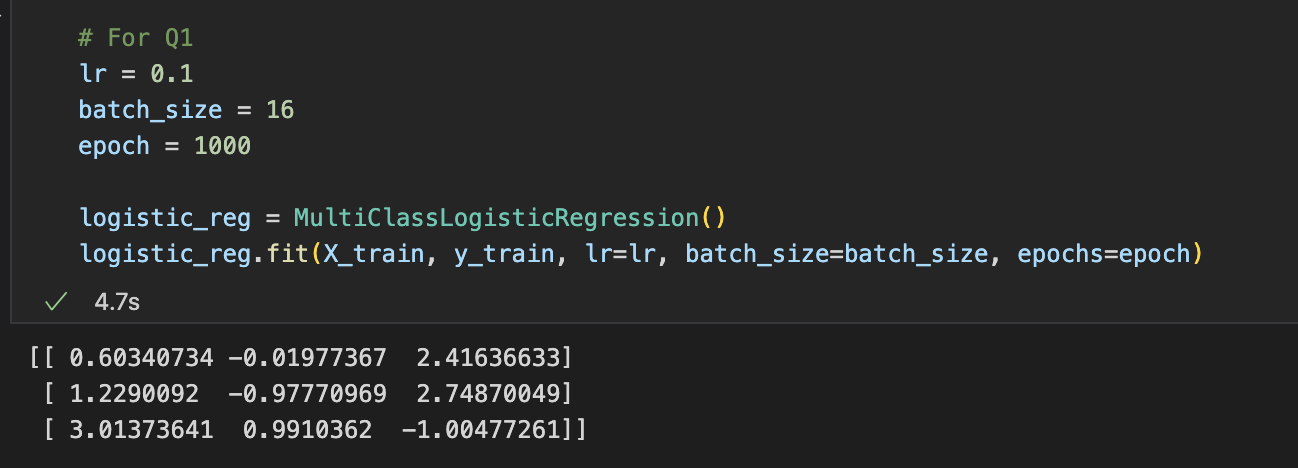
**(20%) Logistic Regression Model**

**Requirements:**

* Use Gradient Descent
* Use CE ([Cross-Entropy](https://zh.wikipedia.org/zh-tw/%E4%BA%A4%E5%8F%89%E7%86%B5)) as your loss function.
* Use [Softmax](https://zh.wikipedia.org/zh-tw/Softmax%E5%87%BD%E6%95%B0) for this multiclass classification task.

**Criteria:**

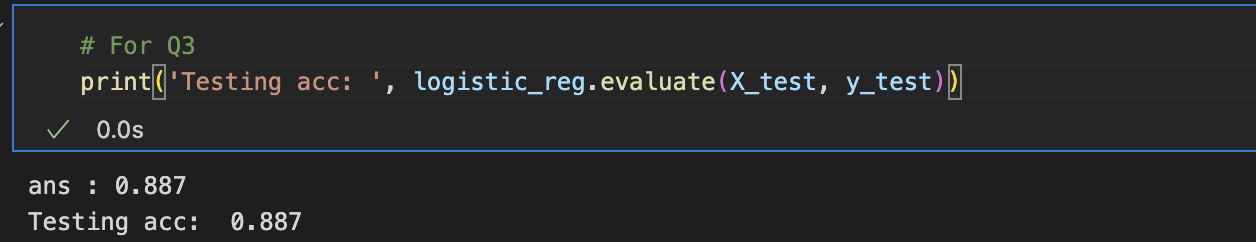
1. (0%) Show the learning rate, epoch, and batch size that you used.



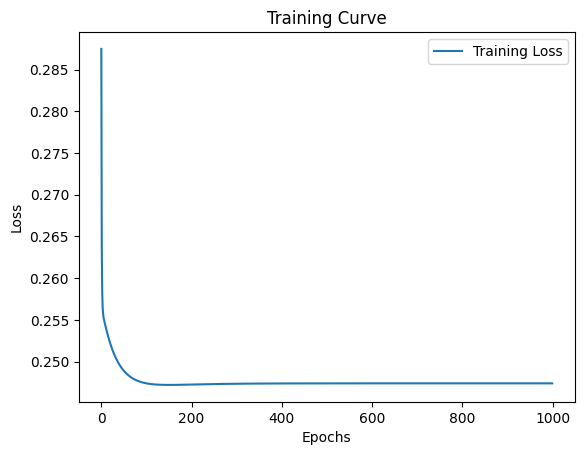
1. (5%) What’s your training accuracy?



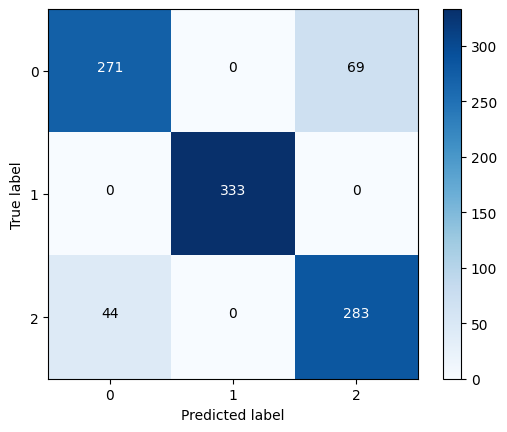
1. (5%) What’s your testing accuracy?



1. (5%) Plot the learning curve of the training. (x-axis=epoch, y-axis=loss)



1. (5%) Show the [confusion matrix](https://ycc.idv.tw/confusion-matrix.html) on testing data.



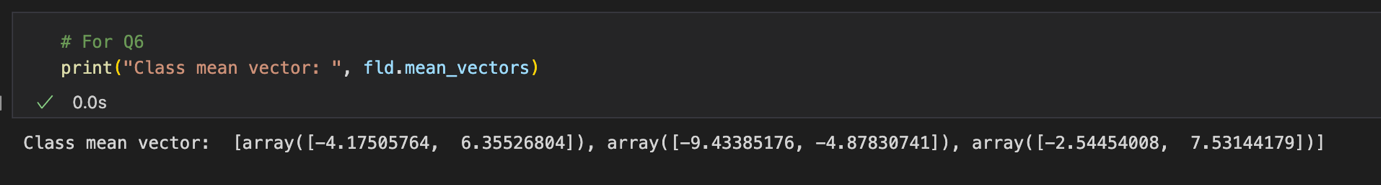
**(30%) Fisher’s Linear Discriminant (FLD) Model**

**Requirements:**

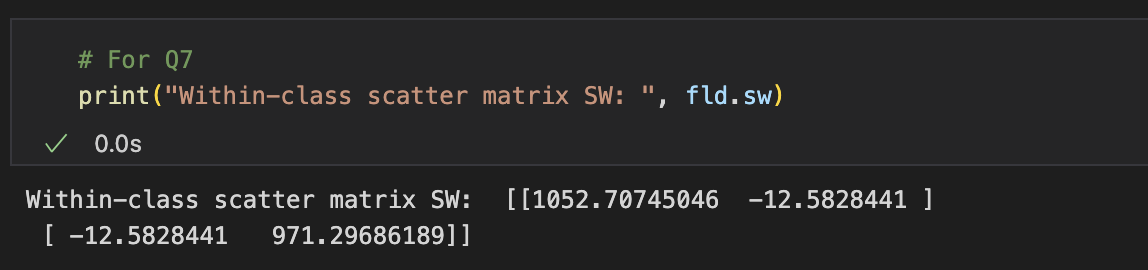
* Use FLD to reduce the dimension of the data from 2 to 1.

**Criteria:**

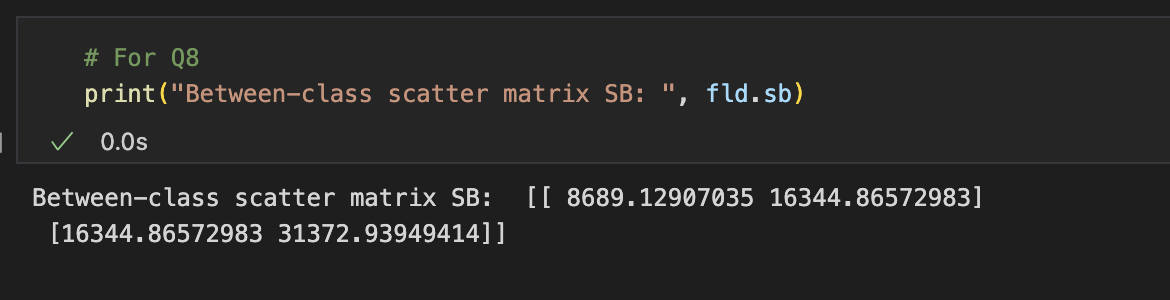
1. (2%) Compute the mean vectors (i=1, 2, 3) of each class on training data.



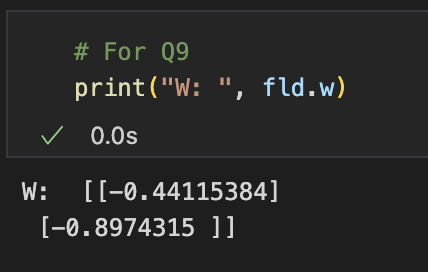
1. (2%) Compute the within-class scatter matrix on training data.



1. (2%) Compute the between-class scatter matrix on training data.

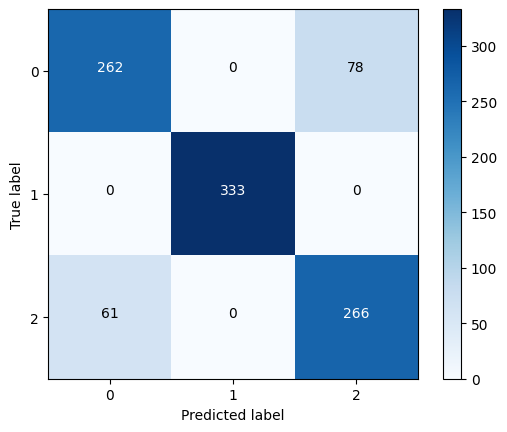


1. (4%) Compute the Fisher’s linear discriminant on training data.

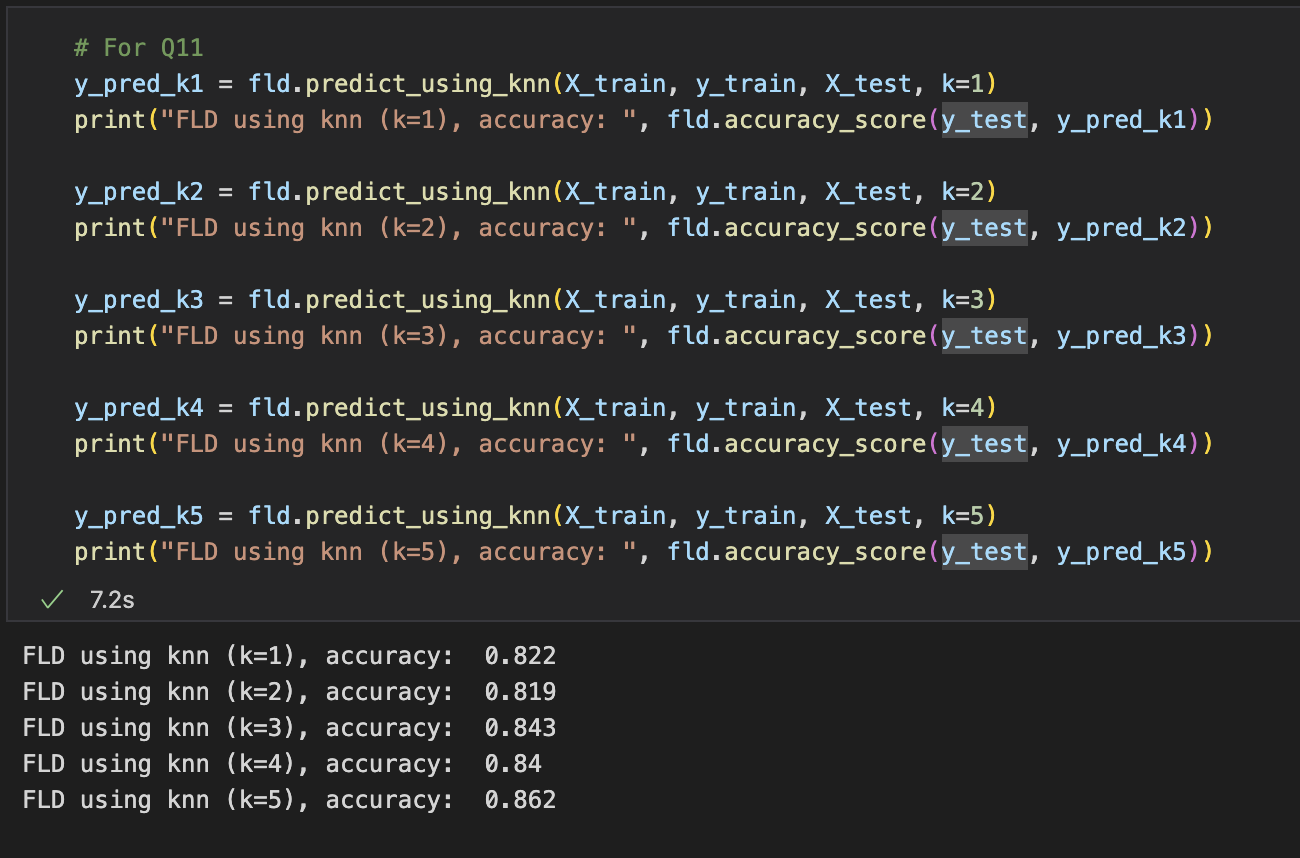


1. (8%) Project the testing data to get the prediction using the shortest distance to the class mean. Report the accuracy score and draw the confusion matrix on testing data.

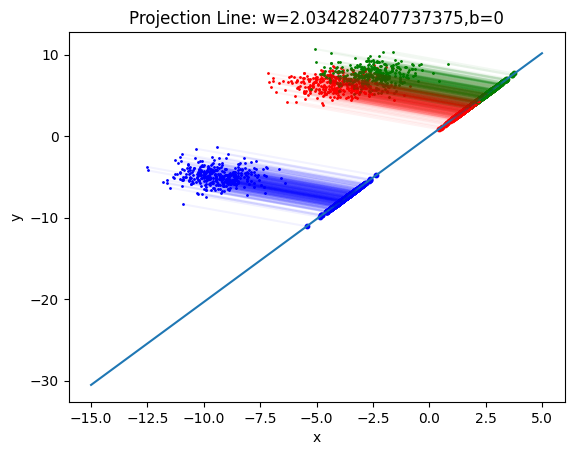
FLD using class mean, accuracy: 0.861



1. (8%) Project the testing data to get the prediction using [K-Nearest-Neighbor](https://www.ibm.com/topics/knn). Compare the accuracy score on the testing data with K values from 1 to 5.



1. (4%)  
   **1)** Plot the best projection line on the training data and show the slope and intercept on the title *(you can choose any value of intercept for better visualization)*  
   **2)** colorize the training data with each class  
   **3)** project all training data points on your projection line. Your result should look like the below image (This image is for reference, not the answer)



**(20%) Train your own model**

**Requirements:**

* Using another dataset that we provided (a real-world dataset).
* Train your model (FLD or Logistics Regression model that you implemented above).
* Try different parameters and feature engineering to beat the baseline.
* Save your testing predictions in the CSV file.

**Criteria:**

1. Explain how you chose your model and what feature processing you have done in detail. Otherwise, no points will be given.

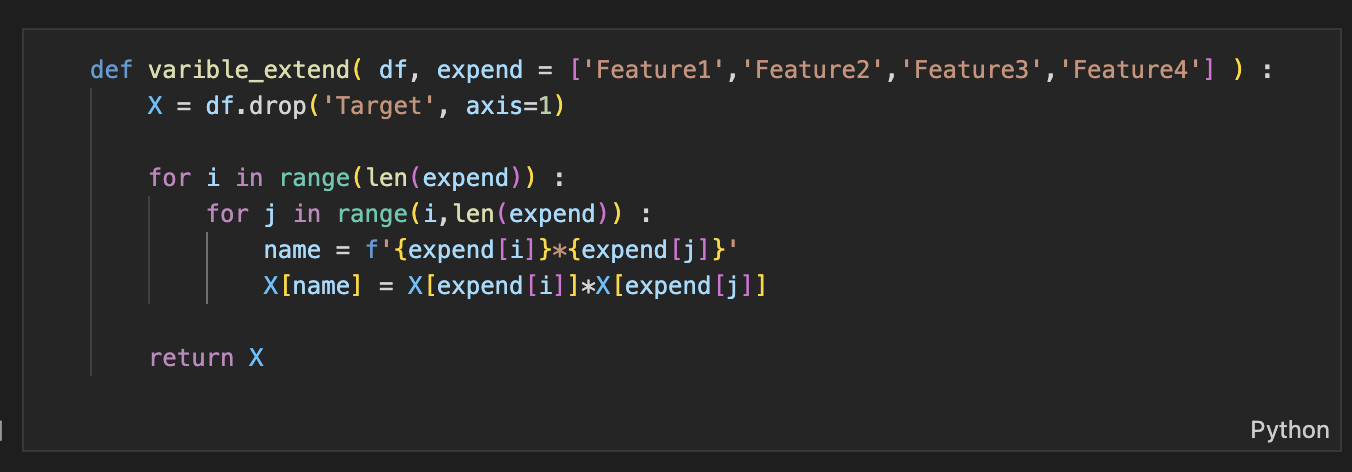
|  |  |
| --- | --- |
| Point | Accuracy |
| 20 | testing acc > 0.921 |
| 15 | 0.91 < testing acc <= 0.921 |
| 8 | 0.9 < testing acc <= 0.91 |
| 0 | testing acc <= 0.9 |

batch\_size: 8

learning rate : 0.9

epoch : 10000000

Use feature :



I used logistic regression as my classifier and did feature expansion because I did not know the effect of feature on this Target, so I left all the expansion features, which can be learned from the last homework that more features are more helpful to the regression performance.

**Part. 2, Questions (30%):**

(6%) 1.Discuss and analyze the performance  
 a) between Q10 and Q11, which approach is more suitable for this dataset. Why?  
 b) between different values of k in Q11. (Which is better, a larger or smaller k?   
 Does this always hold?)

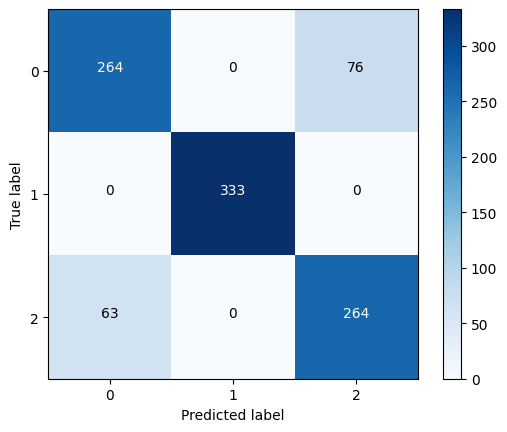
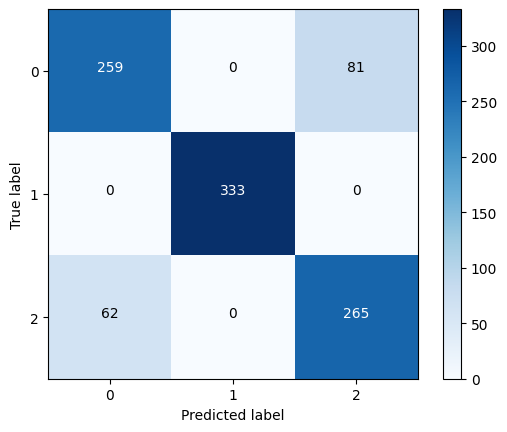
1. **According to Q12, it can be found that two classes are very close to each other,**

**so when using knn, k is set a little larger, it is easy to be biased towards a certain class, but it is not necessarily correct, we can learn from Q11, when k=4 the accuracy rate will be lower than k=3, in this data set should take the class mean to predict better**

**(b)**

****

**The number of this dataset class is 339 334 337, so I tried the numbers in the table above to experiment and found that the larger the number, the better the result.**

**** ****

**k=20 : k=327**

**Because this time the dataset has two classes overlapping each other, so when the k is set a little too large, but will be estimated wrong**

(6%) 2. Compare the sigmoid function and softmax function.

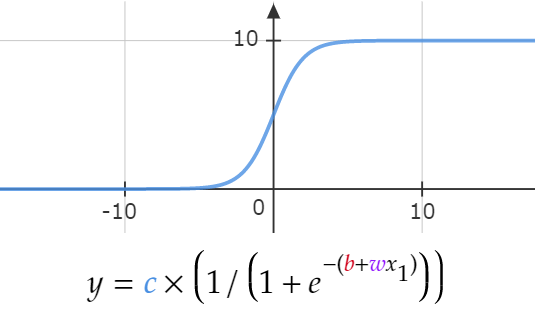
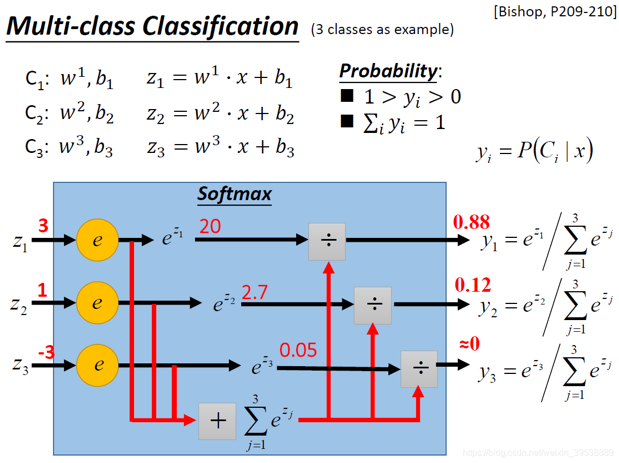
**(1). Range of values:**

**The sigmoid function maps any real-valued number to a value between 0 and 1. This makes it suitable for binary classification problems where the output is either 0 or 1. In contrast, the softmax function maps a vector of real numbers to a probability distribution, meaning that the output values are between 0 and 1 and sum up to 1. This makes it suitable for multiclass classification problems.**

**(2). Shape of the curve:**

**The sigmoid function has an S-shaped curve and is monotonically increasing, which means that**

**the output increases as the input increases. In contrast, the softmax function has a curved shape, and the output is not necessarily monotonically increasing.**

sigmoid softmax

(6%) 3. Why do we use cross entropy for classification tasks and mean square error for

regression tasks?

**Cross entropy for classification tasks:**

**Cross entropy is a loss function that measures the difference between the predicted probability distribution and the actual probability distribution for a classification problem. In other words, it measures the distance between the predicted class probabilities and the true class probabilities. Cross entropy is used in classification tasks because it is more suited to handling probability distributions and penalizes more strongly for larger prediction errors.**

**Mean square error for regression tasks:**

**Mean square error is a loss function that measures the average squared difference between the predicted values and the true values. It is used in regression tasks because it is well-suited for continuous and numeric data. In regression tasks, the goal is to predict a continuous value, such as a real number, and mean square error provides a measure of the average squared difference between the predicted values and the true values.**

**In summary, cross entropy is more suited to classification tasks because it can handle probability distributions and penalizes more strongly for larger prediction errors.**

(6%) 4. In Q13, we provide an imbalanced dataset. Are there any methods to improve Fisher Linear Discriminant's performance in handling such datasets?

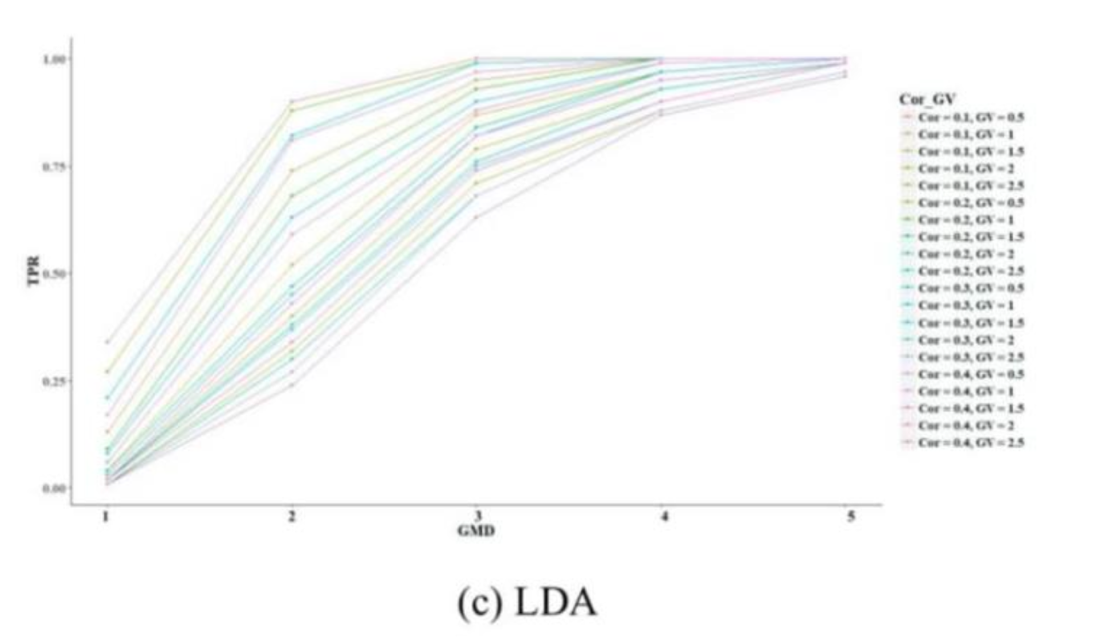
**Yes! Here are a few ways to solve the problem**

**Sampling Techniques :**

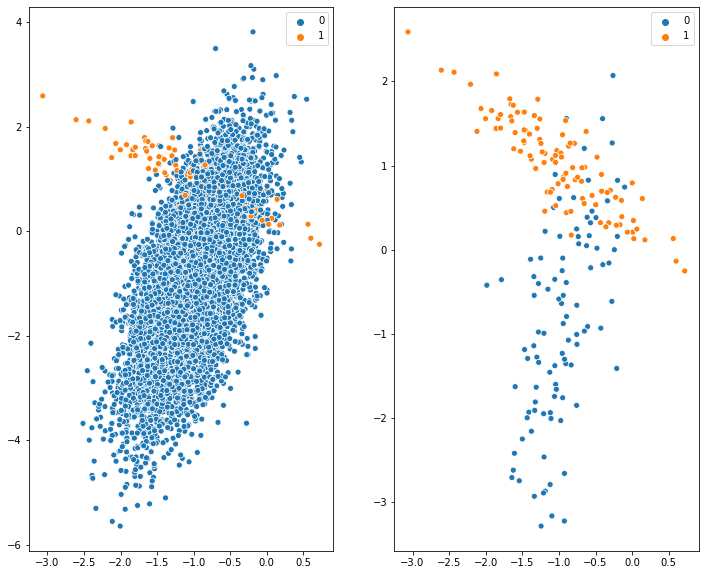
**Sampling techniques are a group of methods used to address imbalanced datasets in machine learning. These techniques aim to balance the distribution of the classes in the dataset by either undersampling the majority class or oversampling the minority class.**

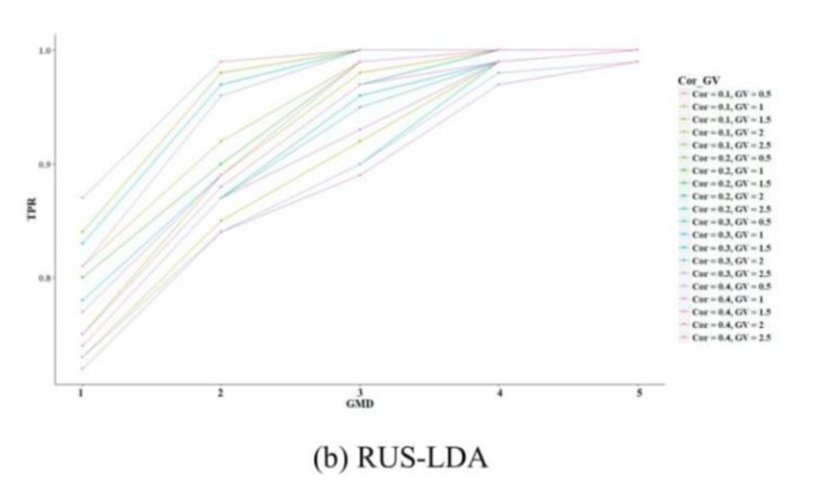
**According to "provide an imbalanced dataset. are there any methods to improve Fisher Linear Discriminant's performance in handling such datasets? "The authors of this paper propose three methods to solve this problem**

LDA :



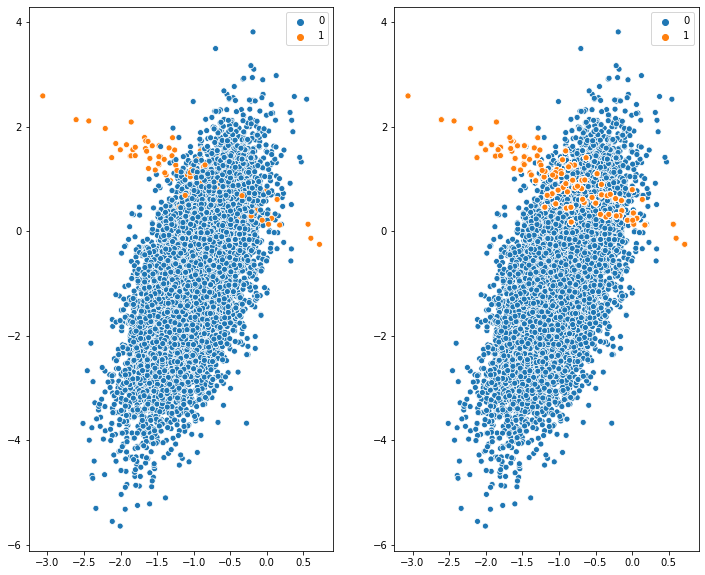
**Random Undersampling: (RUS)**

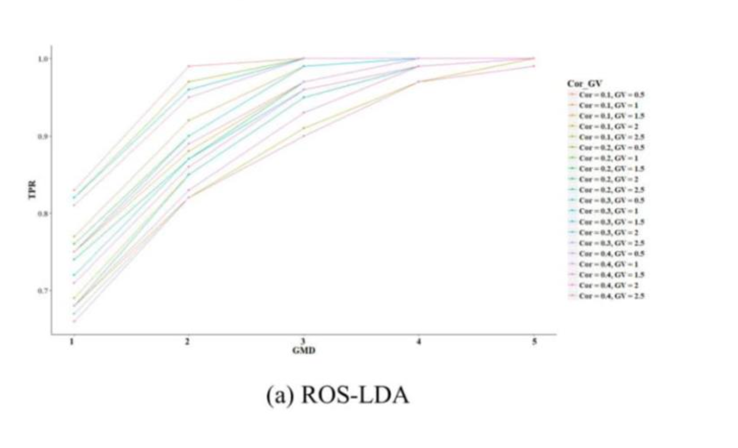
**This technique involves randomly selecting a subset of examples from the majority class to balance the class distribution. While it is simple to implement, it can lead to loss of important information as some examples from the majority class may be discarded.**  Figure : Training data (left), Undersampling (right)



**Random Oversampling: (ROS)**

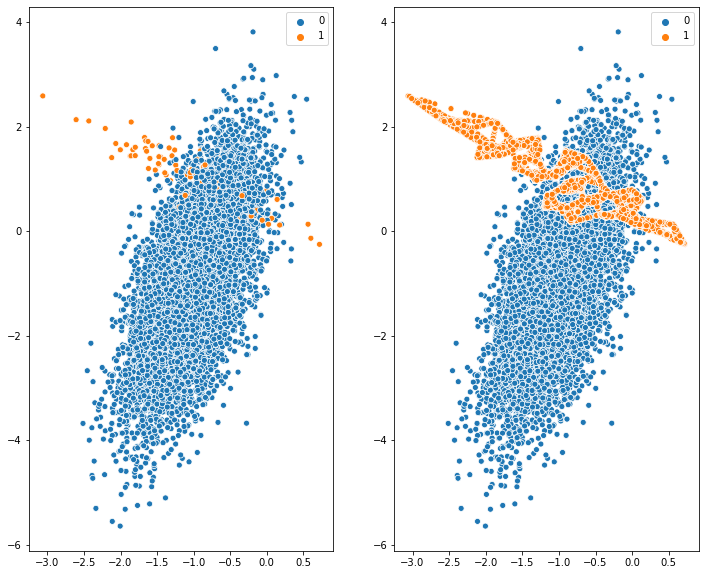
**This technique involves randomly replicating examples from the minority class until the class distribution is balanced. While this method is easy to implement and can prevent loss of information, it can also lead to overfitting and increased computational costs.**

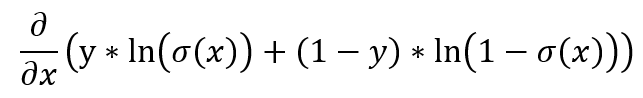
Figure : Training data (left), Oversampling (right)

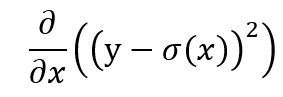


**Synthetic Minority Over-sampling Technique (SMOTE):**

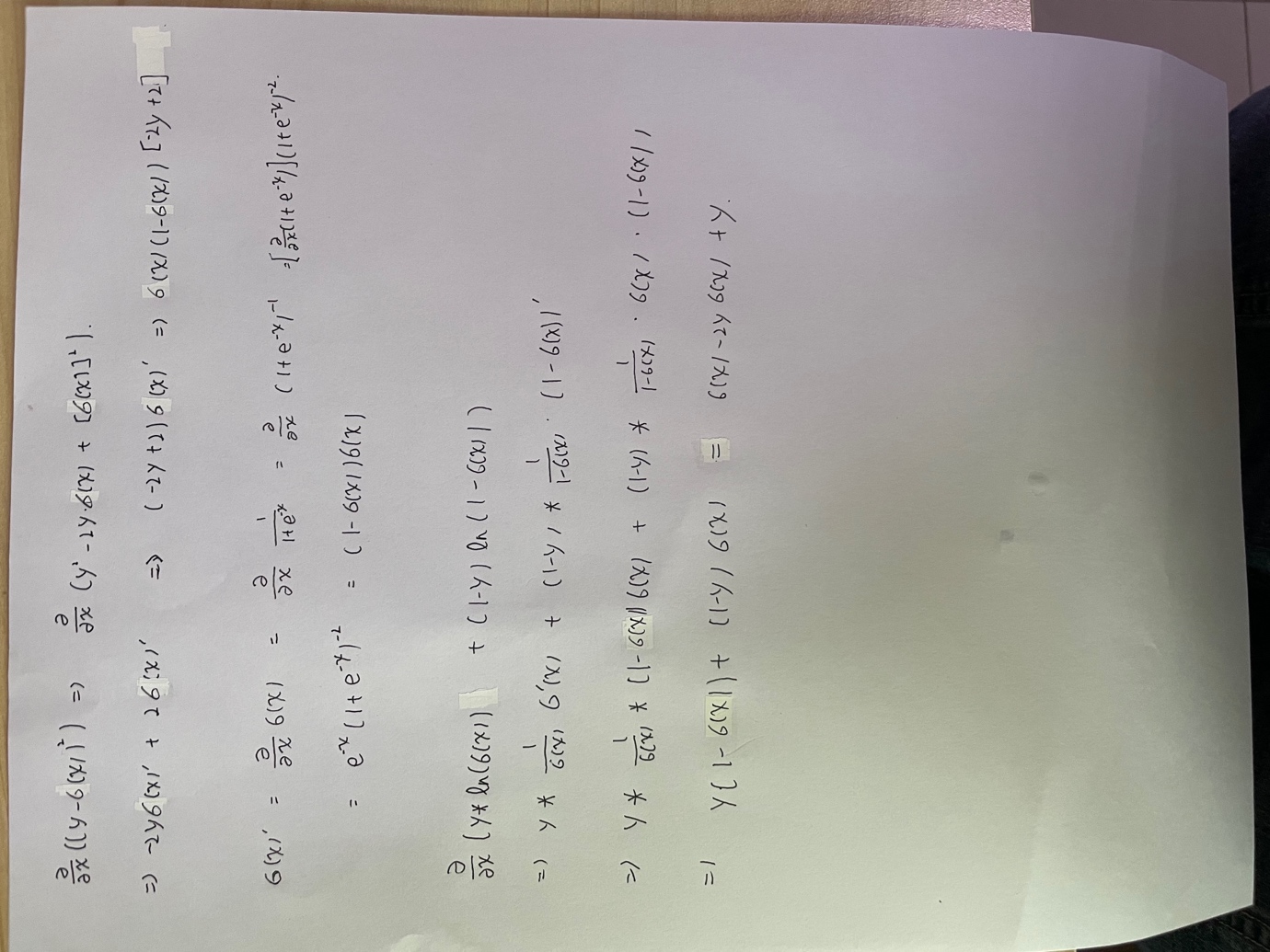
**SMOTE is an oversampling technique that generates synthetic examples for the minority class by interpolating between existing examples. It works by selecting a minority class example and finding its k-nearest neighbors. It then creates synthetic examples by randomly interpolating between the selected example and its neighbors. SMOTE can help to prevent overfitting and can lead to improved generalization.**



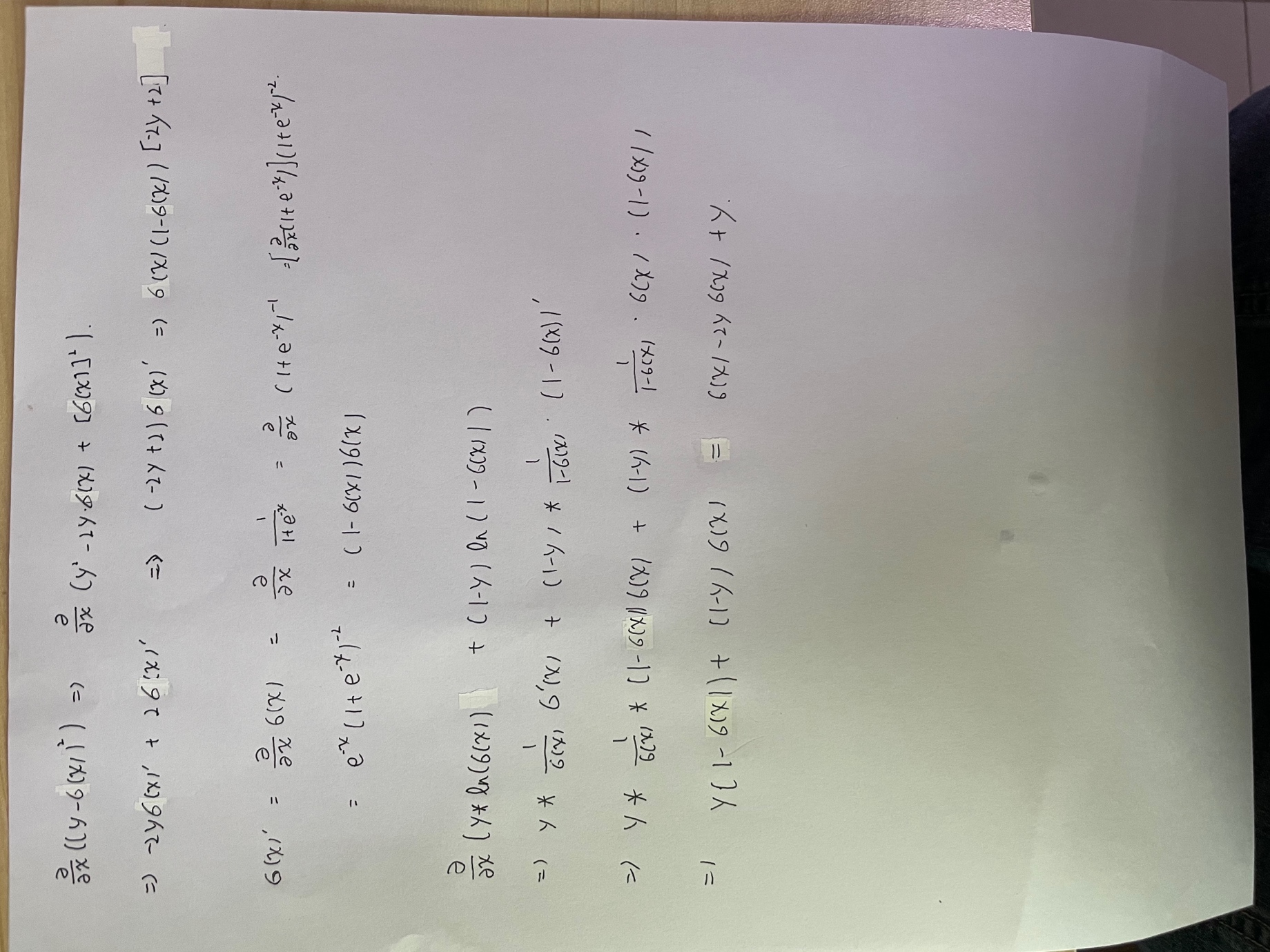
(6%) 5. Calculate the results of the partial derivatives for the following equations. (The first one is binary cross-entropy loss, and the second one is mean square error loss followed by a sigmoid function.)



sigmoid derivatives



(1)



(2)

