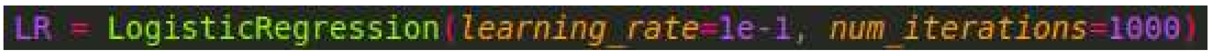
NYCU Pattern Recognition, Homework 2

312551810, **Pauly Alexandre**

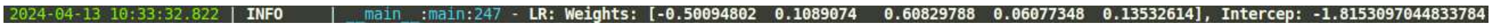
Part. 1, Coding (60%):

(25%) Logistic Regression w/ Gradient Descent Method (slide ref)

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

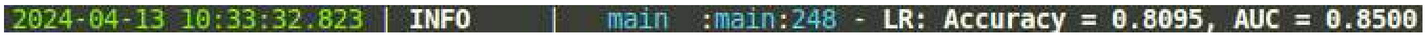


```
LR = LogisticRegression(learning_rate=1e-1, num_iterations=1000)
```



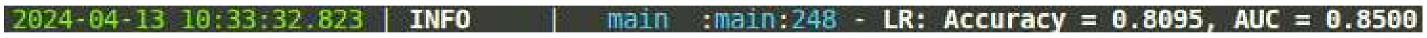
```
2024-04-13 10:33:32.822 | INFO     | __main__:main:247 - LR: Weights: [-0.50094802  0.1089074   0.60829788  0.06077348  0.13532614], Intercep: -1.8153097044833784
```

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



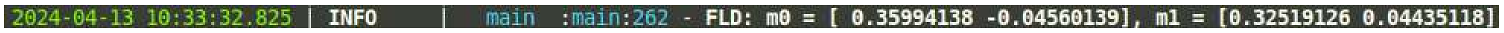
```
2024-04-13 10:33:32.823 | INFO     | __main__:main:248 - LR: Accuracy = 0.8095, AUC = 0.8500
```

3. (5%) Show the AUC of the classification results on the testing set.



```
2024-04-13 10:33:32.823 | INFO     | __main__:main:248 - LR: Accuracy = 0.8095, AUC = 0.8500
```

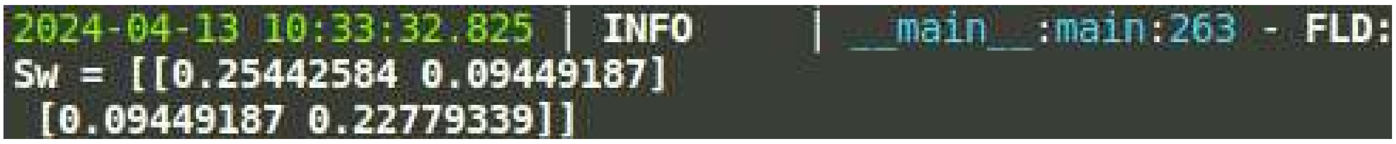
4. (15%) Show the accuracy score of your model on the testing set



```
2024-04-13 10:33:32.825 | INFO     | __main__:main:262 - FLD: m0 = [ 0.35994138 -0.04560139], m1 = [0.32519126 0.04435118]
```

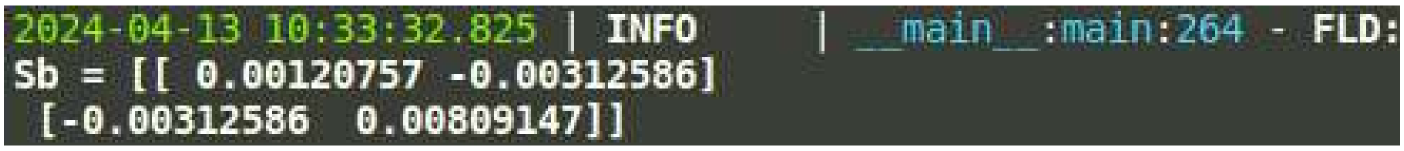
(25%) Fisher Linear Discriminant, FLD (slide_ref)

5. (0%) Show the mean vectors mi (i=0, 1) of each class of the training set.



```
2024-04-13 10:33:32.825 | INFO     | __main__:main:263 - FLD:
Sw = [[0.25442584 0.09449187]
 [0.09449187 0.22779339]]
```

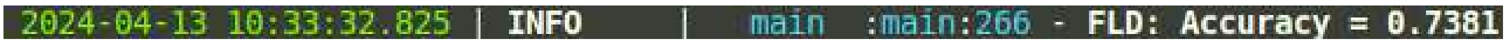
6. (5%) Show the within-class scatter matrix *Sw* and between-class scatter matrix *Sb* of the training set.



```
2024-04-13 10:33:32.825 | INFO     | __main__:main:264 - FLD:
Sb = [[ 0.00120757 -0.00312586]
 [-0.00312586  0.00809147]]
```

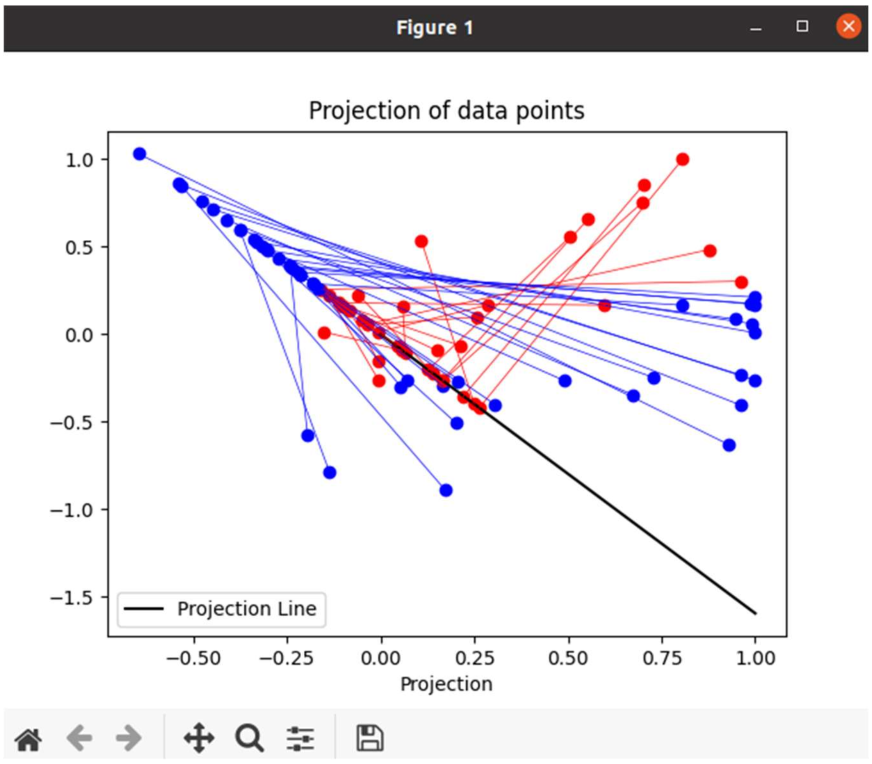
7. (5%) Show the Fisher's linear discriminant *w* of the training set.

8. (15%) Obtain predictions for the testing set by measuring the distance between the



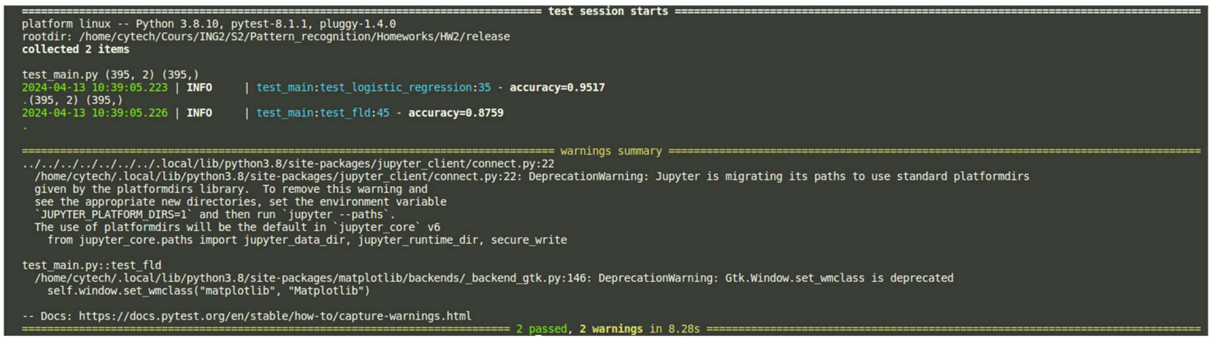
```
2024-04-13 10:33:32.825 | INFO     | __main__:main:266 - FLD: Accuracy = 0.7381
```

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.



(10%) Code Check and Verification

9. (10%) Lint the code and show the PyTest results.



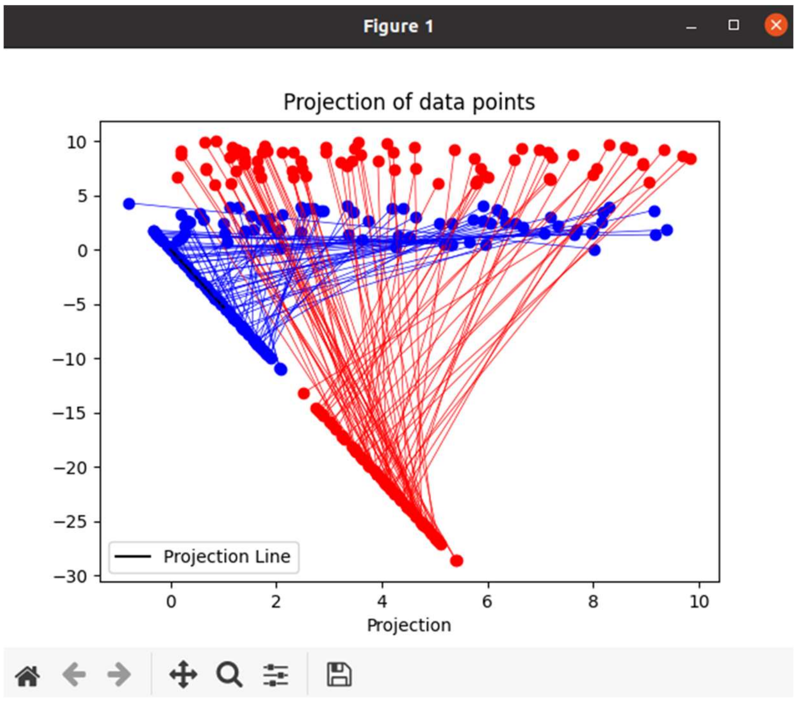
```
=========================================================================== test session starts ============================================================================
platform linux -- Python 3.8.10, pytest-8.1.1, pluggy-1.4.0
rootdir: /home/cytech/Cours/ING2/S2/Pattern_recognition/Homeworks/HW2/release
collected 2 items

test_main.py (395, 2) (395,)
2024-04-13 10:39:05.223 | INFO     | test_main:test_logistic_regression:35 - accuracy=0.9517
.(395, 2) (395,)
2024-04-13 10:39:05.226 | INFO     | test_main:test_fld:45 - accuracy=0.8759
.

============================================================================= warnings summary =============================================================================
../../../../../../../.local/lib/python3.8/site-packages/jupyter_client/connect.py:22
  /home/cytech/.local/lib/python3.8/site-packages/jupyter_client/connect.py:22: DeprecationWarning: Jupyter is migrating its paths to use standard platformdirs
  given by the platformdirs library.  To remove this warning and
  see the appropriate new directories, set the environment variable
  `JUPYTER_PLATFORM_DIRS=1` and then run `jupyter --paths`.
  The use of platformdirs will be the default in `jupyter_core` v6
    from jupyter_core.paths import jupyter_data_dir, jupyter_runtime_dir, secure_write

test_main.py::test_fld
  /home/cytech/.local/lib/python3.8/site-packages/matplotlib/backends/_backend_gtk.py:146: DeprecationWarning: Gtk.Window.set_wmclass is deprecated
    self.window.set_wmclass("matplotlib", "Matplotlib")

-- Docs: https://docs.pytest.org/en/stable/how-to/capture-warnings.html
======================================================================= 2 passed, 2 warnings in 8.28s =======================================================================
```



Part. 2, Questions (40%):

1. (10%) Is it suitable to use Mean Square Error (MSE) as the loss function for Logistic Regression? Please explain in detail.

Logistic regression is a classification model that aims to predict probabilities of belonging to different classes. For this reason, the Mean Squared Error (MSE) loss function does not correctly calculate the loss value for logistic regression because the predicted values are not continuous.

With MSE, the probability predictions will not be properly modeled and will not meet expectations because the target variable is represented as labels (*example: 1 or 2, colors, etc.*). Additionally, since MSE is sensitive to outliers, it assumes a Gaussian distribution of residuals, which may not be the case in a classification context.

To address this issue, it is preferable to use a Cross-Entropy loss function. It is specifically designed for such tasks and measures the divergence between predicted and actual probability distributions. Therefore, it penalizes incorrect predictions more appropriately for classification tasks.

So, for these various reasons, it is more appropriate to use a loss function based on cross-entropy rather than MSE for classification tasks.

2. (15%) In page 31 of the lecture material (linear_classification.pdf), we introduce two methods for performing classification tasks using Fisher's linear discriminator: 1) Determining a threshold, 2) Using the k-NN (k-nearest neighbors) rule. Please discuss at least three aspects, either advantages or disadvantages, of using the k-NN method compared to determining a threshold (resources, performance, etc.).

When comparing the use of the k-nearest neighbors (k-NN) method to that of threshold determination in the context of Fisher's linear discriminant, several aspects can be considered:

Complexity and Resources:

- Advantage: The calculation of classification with the k-NN method is simple and fast. It does not require any learning phase as the modeling is done directly on the entire dataset. This is advantageous in terms of resources and especially time.
- Disadvantage: One of the main disadvantages of k-NN is that it requires a lot of memory and computational resources to store and process the entire training data. As the size and dimensionality of the data increase, calculating distance and finding the nearest neighbors becomes costlier and more time-consuming.

Performance:

- Advantage: It is a learning algorithm that does not build any explicit model. Instead, it stores the entire training data and only performs calculations when a new query is presented. This makes k-NN adaptable to changes in data and capable of handling noisy and nonlinear data. Additionally, k-NN makes no assumptions about the distribution or underlying structure of the data.
- Disadvantage: Dependency on the choice of k and the distance metric. These two parameters can have a significant impact on the performance and accuracy of k-NN. If k is too small, k-

NN can be influenced by noise and outliers, while if k is too large, k-NN can lose local information and class boundaries. Therefore, it is important to experiment with different values of k and different distance measures to find the optimal values for the data.

Interpretability:

- Advantage: The simple demonstration from a plot highlights the different classes created by k-NN, making it easily interpretable by anyone.
- Disadvantage: A final limitation of k-NN is that it may struggle with imbalanced classes and missing values in the data. Imbalanced classes occur when some classes have many more instances than others, which can bias k-NN towards the majority class and ignore the minority class. This necessitates either resampling, weighting, or balancing the classes.

3. (15%) In logistic regression, what is the relationship between the sigmoid function and the softmax function? In what scenarios will the two functions be used respectively?

The sigmoid function and the softmax function are both commonly used activation functions in machine learning, particularly in logistic regression and neural networks for classification tasks.

Relationship between the Sigmoid and Softmax functions:

- The sigmoid function is defined as the next picture where x is the input to the function. It maps any real number to the interval (0,1), making it suitable for binary classification problems. It is used to model the probability that a given input belongs to the positive class.

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

- The softmax function is a generalization of the sigmoid function and is commonly used for multi-class classification problems. It is defined as the next picture where x_i is the input for class i and K is the total number of classes. It computes the probability distribution over K different classes, ensuring that the sum of probabilities across all classes is equal to 1.

  $$s(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}$$

Therefore, the sigmoid function is a special case of the softmax function when there are only two classes (binary classification).

Using for each function:

- **Sigmoid function:** It is generally used in binary classification tasks where the output is binary (0 or 1). For example, in logistic regression, the sigmoid function is applied to the linear combination of features to model the probability that an instance belongs to the positive class.
- **Softmax function:** It is used in multi-class classification tasks where there are more than two classes. For instance, in neural networks, the softmax function is often employed in the output layer to generate a probability distribution over multiple classis. Each output neuron represents the probability of the corresponding class.

In summary, while the sigmoid function is used for binary classification problems, the softmax function is utilized for multi-class classification problems to produce a probability distribution over multiple classis.