



# 2024 Spring Pattern Recognition Homework 2 Announcement

Release Date: 2024/04/10 12:00

# Homework 2

- Deadline: 23:59, Apr. 24th (Wed), 2024
- **Coding** (60%): Implement linear classification methods by **only** using *numpy*.
  - Submit your code in executable python files (.py).
  - Report the outcome and parameters by screenshots to the questions.
- **Handwritten Questions** (40%): Answer questions about linear classification.
  - Answer the questions in the report.
  - You must use the template and in digital-typed (no handwritten scan)
  - In English

# Links

- [Questions and Report template](#)
- [Sample code / Dataset](#)

# Coding Environment

- Recommendation: Python 3.9 or higher
- Tips
  - We recommend you to use **virtual environments** when implementing your homework assignments.
  - Here are some popular virtual environment management tools
    - [Poetry](#)
    - [Conda](#)
    - [Virtualenv](#)

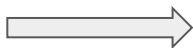


# Numpy

- High efficient vector and matrix operations
- Numpy Tutorial: [Link](#)

element-wise  
multiply

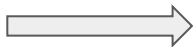
```
a = np.array([1, 2, 3])  
b = np.array([4, 5, 6])  
for i in range(a.shape[0]):  
    a[i] *= b[i]  
print(a)  
# a = [ 4 10 18]
```



```
a = np.array([1, 2, 3])  
b = np.array([4, 5, 6])  
a *= b  
print(a)  
# a = [ 4 10 18]
```

square root

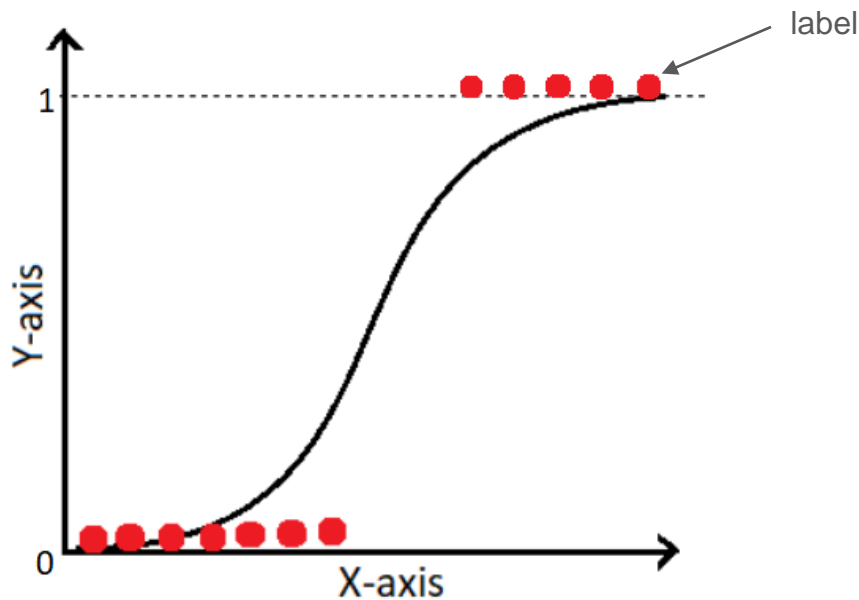
```
import math  
a = np.array([1, 4, 9])  
for i in range(a.shape[0]):  
    a[i] = math.sqrt(a[i])  
print(a)  
# a = [1 2 3]
```



```
a = np.array([1, 4, 9])  
a = np.sqrt(a)  
print(a)  
# a = [1 2 3]
```

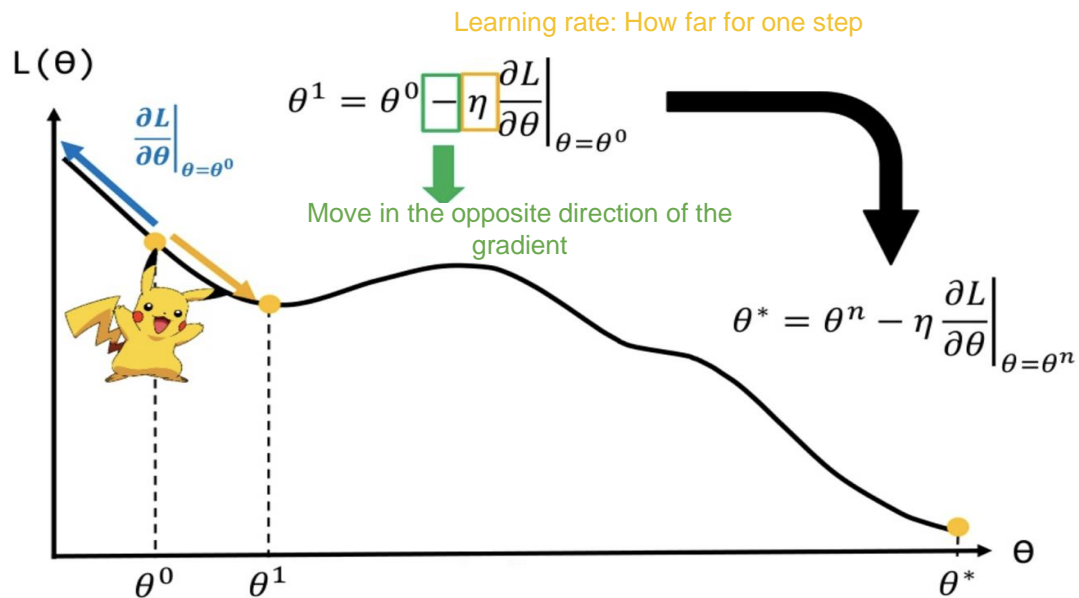
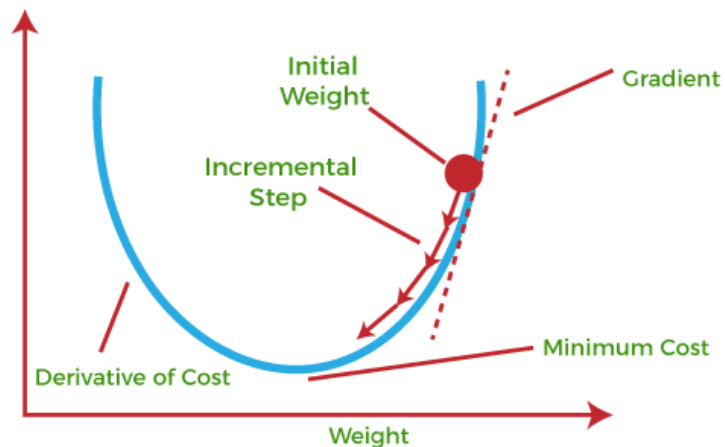
# Logistic Regression for Binary classification

- Find the best value of the weights and the intercept of a logistic model



# Logistic Regression

- Gradient Descent

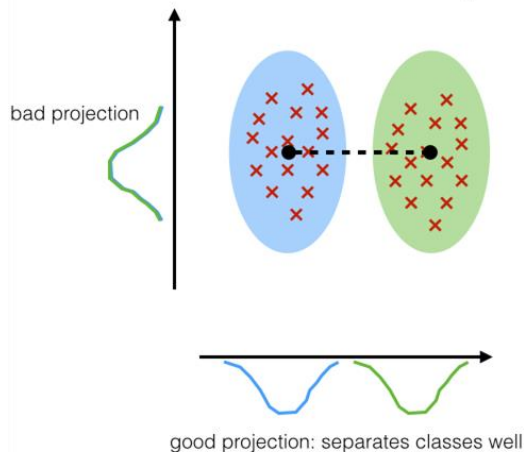


# Fisher's Linear Discriminant, FLD

- FLD seeks the projection  $w$  that gives a large distance between the projected data center (means) while giving a small variance within each class.

## LDA:

maximizing the component axes for class-separation




$$J(\mathbf{W}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2}$$

Between-class variance

Within-class variance



# Dataset and Environment

- Sonar dataset - A binary classification problem
- Datapoints
  - 208 (166 for train / 42 for test)
- Features
  - 60 features (already normalized) 
- Target
  - Rock or Mine (0 / 1)
- Required packages: ``numpy``, ``pandas``, ``matplotlib``, ``loguru``, ``flake8``, ``pytest``,  
``scikit-learn``

# Logistic Regression (25%)

- Requirements

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

- Grading Criteria

- (0%) Show the hyperparameters (learning rate and iteration, etc) that you used. (-3 pts if absent)
- (5%) Show your weights and intercept of your model.
- (5%) Show the [AUC](#) score of the classification results on the testing set.
  - **sklearn is allowed to compute AUC score (only for this)**
- (15%) Show the **accuracy score** of your model **on the testing set**.

Accuracy	Score (pt)
$\geq 0.8$	15 pt
$\geq 0.75, < 0.8$	10 pt
$\geq 0.7, < 0.75$	5 pt
$< 0.7$	0 pt

# Fisher's Linear Discriminant, FLD (25%)

- Requirements:
  - Implement FLD to project the data from 2-dimensional to 1-dimensional space.
- Criteria:
  - (0%) Show **the mean vectors  $m_i$  ( $i=0, 1$ )** of each class of the training set. (-3 pts if absent)
  - (5%) Show the **within-class scatter matrix  $S_W$**  and **between-class scatter matrix  $S_B$**  of the training set.
  - (5%) Show the Fisher's linear discriminant  $w$  of the training set.

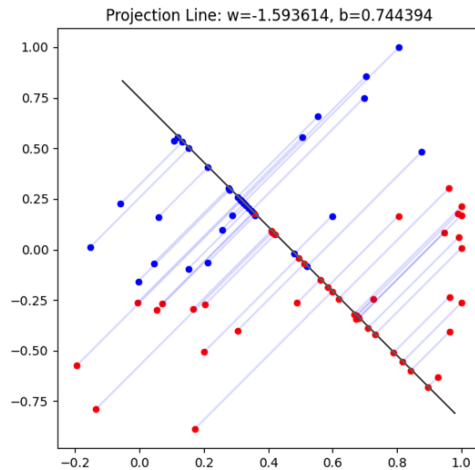
# Fisher's Linear Discriminant

Accuracy	Score (pt)
$\geq 0.70$	10 pt
$\geq 0.6, < 0.7$	5 pt
$< 0.6$	0 pt

- Criteria

- (15%) 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.
  - **Plot the projection line** trained on the training set and show the slope and intercept on the title.
  - **Obtain the prediction of the testing set**, plot and **colorize** them based on the prediction.
  - Project all testing data points onto 1d space.
- Also, **Show the accuracy score** on the testing set.

**Just an example**



# Code Output

- Do not modify the main function architecture heavily.
- Your code output will look like this

```
2024-04-06 18:31:57.199 | INFO | __main__:main:169 - LR: Weights: [REDACTED], Intercep: [REDACTED]
2024-04-06 18:31:57.200 | INFO | __main__:main:170 - LR: Accuracy=0.8095, AUC=0.8477
2024-04-06 18:31:57.203 | INFO | __main__:main:183 - FLD: m0=[REDACTED], m1=[REDACTED]
2024-04-06 18:31:57.203 | INFO | __main__:main:184 - FLD:
Sw=
[REDACTED]
2024-04-06 18:31:57.203 | INFO | __main__:main:185 - FLD:
Sb=
[REDACTED]
2024-04-06 18:31:57.204 | INFO | __main__:main:186 - FLD:
w=
[REDACTED]
2024-04-06 18:31:57.204 | INFO | __main__:main:187 - FLD: Accuracy=0.7381
```

1. [PEP8](#)
2. [Google Python Style](#)

# Additional Requirements

Code Check and Verification: **Lint** the code and show the **PyTest** results (10%)

- Code linting: `$ flake8 main.py`
  - **-2pt** per warning / error
- Run PyTest: `$ pytest ./test_main.py -s`
  - **-5pt** per failed case

```
./main.py:103:1: W391 blank line at end of file
1      W391 blank line at end of file
```

```
===== test session starts =====
platform linux -- Python 3.9.16, pytest-8.1.1, pluggy-1.4.0
rootdir: /home/seanyu/lectures/nycu/ml-and-pattern-recognition/hw2
configfile: pyproject.toml
collected 2 items

test_main.py (395, 2) (395,)
2024-04-06 18:56:48.208 | INFO | test_main:test_logistic_regression:35 - accuracy=0.9517
.(395, 2) (395,)
2024-04-06 18:56:48.213 | INFO | test_main:test_fld:45 - accuracy=0.8759
.

===== 2 passed in 5.08s =====
```

# Handwritten Questions (40%)

## 2-1 (10%)

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

## 2-2 (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.).

## 2-3 (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?

# Report

- Please follow the report template format. (-5pts if not use the template)
- [Link](#)



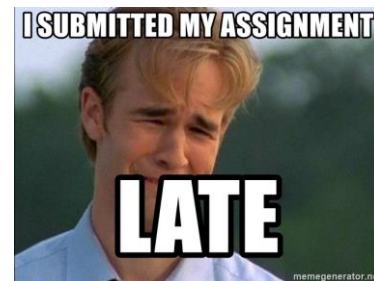
# Submission

- Compress your **code** and **report** into a **.zip file** and submit it to E3.
- Report should be written in English. (-5 pts if not English)
- <STUDENT ID>\_HW2.zip
  - main.py
  - setup.cfg
  - test\_main.py
  - <STUDENT ID>\_HW2.**pdf** (NO .doc, .docx or others format)
- Don't put the data (e.g. train.csv / test.csv) into submission file

# Other rules

- **Late Policy**: A penalty of **20 points** per additional late day. (-20pt / delayed.day)

- For example, If you get 90 points but delay for two days, your will get only 50 points!



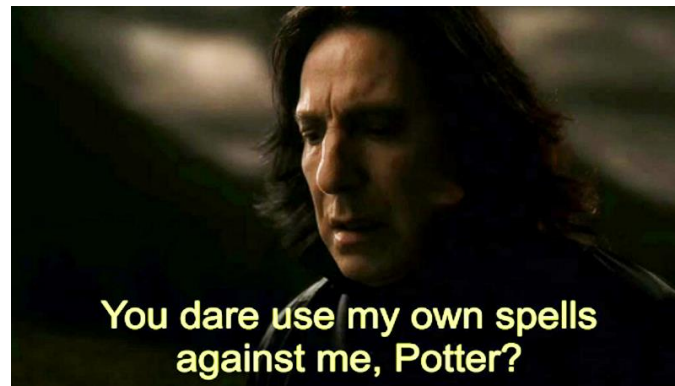
- **No Plagiarism**: You should complete the assignment by yourself. Students engaged in plagiarism will be penalized heavily. Super serious penalty.

- e.g. -100pt for the assignment or failed this course, etc
- Report to academic integrity office



# AI-Assistant

- Not recommended but no forbidden
- **Copy-and-Paste answers from the AI-Assiant will be seen as Plagiarism**
  - However, you can have your own answer first then rephrase it by AI-Assiant.
- Some questions might be parts of final exam, make sure you understand the concept



# FAQs

- If you have other questions, ask on **E3 forum** first! We will reply as soon as possible.
  - If the E3 discussion area still have issues, feel free to write email to TAs (And remember to cc all TAs).

Have Fun!