
# Pattern Recognition
# Homework 2 Announcement

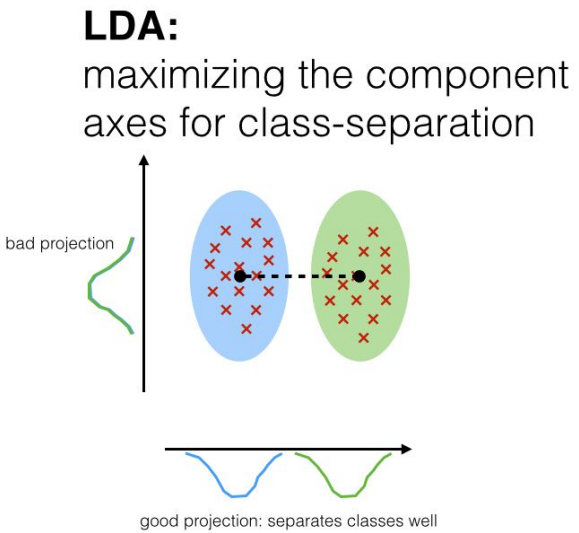
Lastest update: 2023.03.22 11:50

# Homework 2

- Deadline: Apr. 05, Wed. at 23:59
  - Code assignment (70%)
    - Implement Logistic Regression and Fisher's Linear Discriminant using only NumPy.
  - Questions (30%)
    - Write your answer in detail in the report.
- Question: Link
- Sample code: Link
- Dataset: Link

# Fisher's Linear Discriminant

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




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

*Between-class variance*

*Within-class variance*

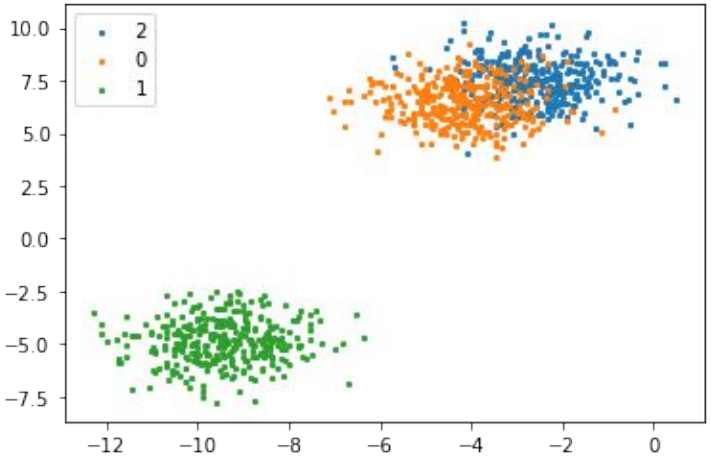
https://sebastianraschka.com/Articles/2014_python_lda.html, https://sthalles.github.io/fisher-linear-discriminant/

# Blob Dataset (for Q1~Q12)

- sklearn.datasets.make_blobs
- 2 features, 3 labels
- Training set
- Testing set (with label)

PR_HW2_blob_test.csv

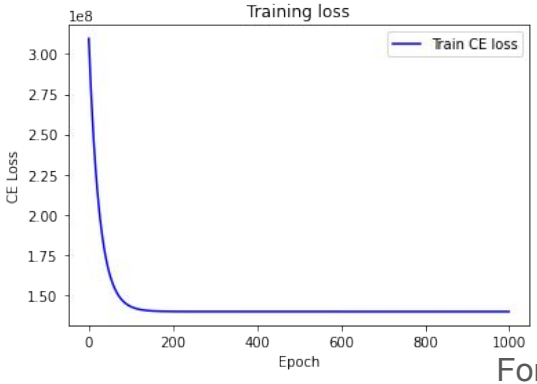
PR_HW2_blob_train.csv

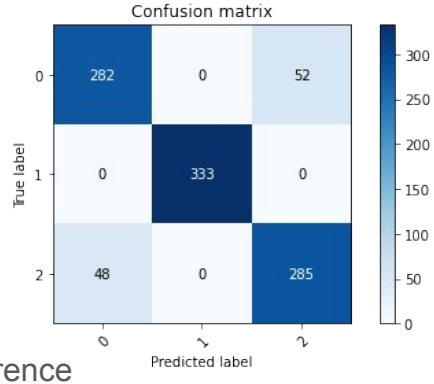



- Train the models using the training set and evaluate using the testing set by yourself.

# Logistic Regression Model - 20%

- Like HW1, use only Numpy to implement the Logistic Regression.
- Tune the parameters and use gradient descent methods to train your model.
- Cross Entropy and Softmax.
- Your model should get at least 0.9 accuracy score on testing set.
- Plot the learning curve and the confusion matrix.




For your reference

# Fisher’s Linear Discriminant (FLD) Model - 30%

- Use only Numpy to implement the FLD.
- Show the following results:
  - Mean vectors $\mathbf{m_i}$
  - Within-class scatter matrix $\mathbf{s_W}$
  - Between-class scatter matrix $\mathbf{s_B}$
  - Fisher’s linear discriminant **w**

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

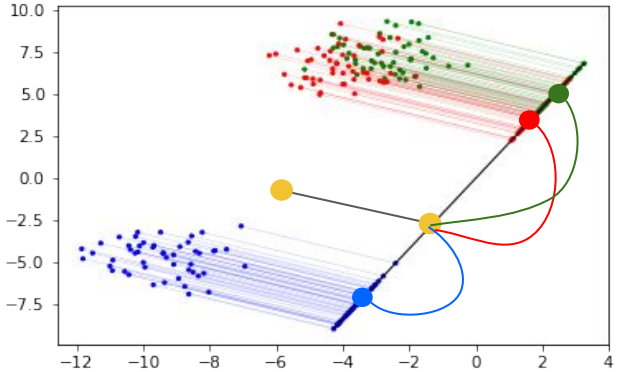
*Between-class variance*

*Within-class variance*

# Fisher’s Linear Discriminant (FLD) Model - 30%

- Predict the testing data.
  - Determined by the shortest distances to the class mean.
  - Determined by KNN (k=1, 2, 3, 4, 5)

    (Refer to chapter 3 slide, page 30)




- Analyze the performance between the two methods and also the different values of k.

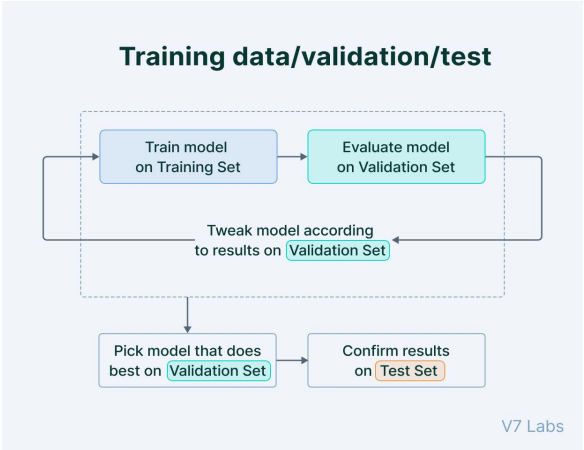
# Train your own model (20%)

- A real word dataset
  - training set
  - validation set
  - testing set
- 4 features, 3 labels
- Imbalanced data

PR_HW2_test.csv

PR_HW2_train.csv

PR_HW2_val.csv




```
1  df_train.head()
```

| | Feature1 | Feature2 | Feature3 | Feature4 | Target |
|---|---|---|---|---|---|
| 0 | 0.00668 | 0.00192 | 0.682 | 0.996 | 2.0 |
| 1 | 0.00680 | 0.00106 | 0.503 | 0.996 | 1.0 |
| 2 | 0.00742 | 0.00106 | 0.482 | 0.991 | 1.0 |
| 3 | 0.00685 | 0.00178 | 0.650 | 0.998 | 2.0 |
| 4 | 0.00680 | 0.00163 | 0.623 | 0.996 | 2.0 |

```
1  df_test.head()
```

| | Feature1 | Feature2 | Feature3 | Feature4 | Target |
|---|---|---|---|---|---|
| 0 | 0.00699 | 0.000877 | 0.451 | 0.994 | NaN |
| 1 | 0.00736 | 0.001370 | 0.549 | 0.998 | NaN |
| 2 | 0.00687 | 0.001420 | 0.580 | 0.992 | NaN |
| 3 | 0.00752 | 0.002520 | 0.737 | 0.996 | NaN |
| 4 | 0.00685 | 0.000910 | 0.464 | 0.992 | NaN |

https://www.v7labs.com/blog/train-validation-test-set

# Train your own model (20%)

- You can only use the FLD/Logistic Regression that you implemented.
- You can try different learning rates, epochs, batch-size, and features to beat the baseline.
- Explain in detail how you choose the model, parameters, and features in the report. Otherwise, extra penalty.
- Some hints in HW1 may still be helpful in HW2.
- Predict the testing data and save the result into a CSV file.

# Train your own model (20%)

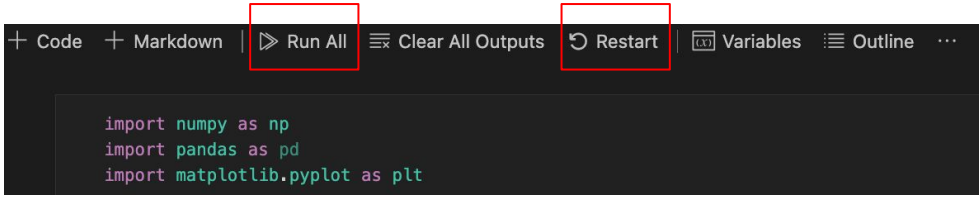
- Evaluation is based on testing accuracy.
- Testing data distribution is guaranteed to be similar to validation data.

| Point | Testing 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 |

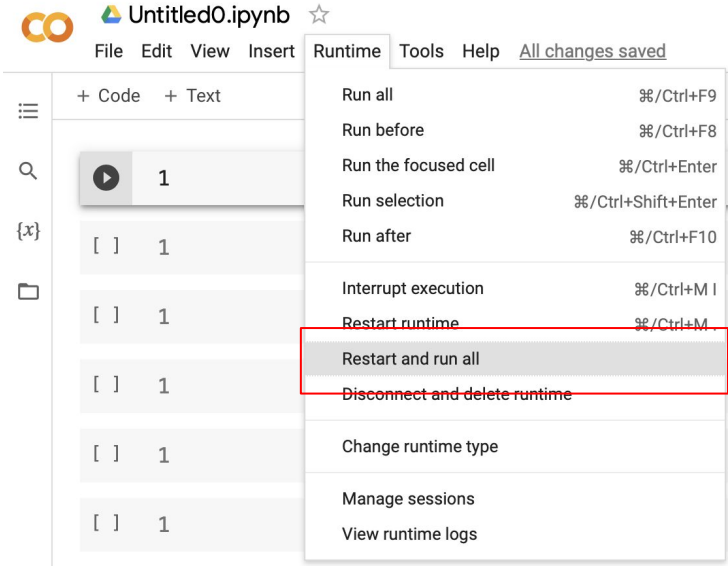
# Report

- Please write your report in English.
- Please follow the HW1 report template.
- You must type the answer and also screenshot at the same time for the coding part.
- **Answer each question as clearly as possible**. You will get an extra penalty for only the brief answer.

# Submission




- Compress your .ipynb, .pdf, and .csv into a zip file and submit it on E3.
- Before submission:
  - Restart and run All
  - Save and submit the .ipynb (keep all cell outputs)
  - Get 0 points if you do not keep the cell outputs.
- <STUDENT ID>_HW2.zip
  - <STUDENT ID>_HW2.ipynb
  - <STUDENT ID>_HW2.pdf
  - <STUDENT ID>_prediction.csv




```
❯ zip -r 310551056_HW1.zip 310551056_HW1.ipynb 310551056_HW1.pdf 310551056_prediction.csv
  adding: 310551056_HW1.ipynb (deflated 34%)
  adding: 310551056_HW1.pdf (deflated 8%)
  adding: 310551056_prediction.csv (deflated 57%)
```

For your reference

# Late policy

- We will deduct a late penalty of 20 points per additional late day
- For example, If you get 90 points but delay for two days, you will get only 90-(20 x 2) = 50 points!