

# 2024 Data Mining

---

HW3

# Task introduction

- Anomaly Detection

- TA use the Letter Image Data features and select 6 letters to form the training set, and randomly add some other 4 letters as outliers in testing set.
- Implement **machine/deep learning model** to do anomaly detection.
- **All package is available (sklearn, keras, pytorch etc.).**
- **Do not use any pretrained model & Do not use any extra data for training**, but it is acceptable to use such data for validation purposes.

- Requirement

- Upload your submission to Kaggle
- Submit a report and your source code to E3

- **Deadline is 6/21 23:59, no late submission**



# Dataset

## UCI Letter Image Recognition Data Set.

- training.csv
  - Randomly sampled 6 letters(each label is 700) from Letter Image Recognition Data Set.
  - [link](#)
- test\_X.csv
  - Randomly sample 600 from previous letters, and randomly select 400 other letters.
  - Please use the features to assign weight values to indicate whether each letter is an outlier or not.
  - [link](#)

# Training Data

letter	x-box	y-box	width	high	onpix	x-bar	y-bar	x2bar	y2bar	xybar	x2ybr	xy2br	x-egx	yegvy	y-egx	yegvx	
B	4	8	6	6	6	5	9	7	4	6	10	5	6	2	8	6	10
B	1	0	1	0	0	0	7	7	6	4	7	6	7	1	8	5	9
B	4	8	6	6	6	8	8	8	4	3	6	7	7	6	11	8	9
B	4	7	6	5	5	5	8	6	5	6	9	6	7	3	8	7	9
B	9	14	7	8	5	6	8	8	5	7	10	6	8	6	6	7	9
B	4	7	6	5	5	5	7	8	5	5	9	6	6	3	7	6	7
B	4	7	5	5	6	8	7	7	6	6	6	6	6	2	8	6	9
B	4	7	5	5	4	8	6	6	4	6	9	5	6	2	8	6	10
B	5	10	6	8	7	9	7	7	3	5	7	6	8	7	8	6	9
B	4	9	4	7	4	6	6	7	9	7	7	6	7	2	8	9	10
B	3	7	5	5	5	8	8	8	4	5	7	5	6	4	8	5	8
B	4	10	6	8	7	8	7	7	4	7	6	6	6	6	8	6	10
B	6	9	8	7	7	7	7	8	6	5	6	5	6	4	9	7	6
B	4	10	5	8	7	6	6	8	8	5	7	5	7	2	8	7	9
B	2	3	4	2	2	8	7	7	3	5	10	5	6	2	8	4	9
B	6	9	8	7	6	10	6	6	3	7	10	3	7	6	8	7	11
B	6	8	9	7	9	7	7	8	5	5	8	6	8	7	7	9	6
B	4	10	5	7	7	7	8	7	6	5	7	6	6	6	8	6	10
B	3	6	4	4	5	7	7	7	6	5	7	6	6	2	8	6	10
B	7	11	9	8	8	8	10	6	3	6	10	3	7	5	7	6	11
B	4	9	6	6	6	6	6	8	7	4	7	5	6	4	8	5	6
B	4	9	4	7	5	6	8	8	8	6	7	5	7	2	8	7	9
B	3	7	3	5	3	6	8	8	8	6	7	5	7	2	8	9	10
B	2	6	4	4	3	8	8	8	5	7	7	6	6	2	8	6	9

All features are given.

# Testing Data

x-box	y-box	width	high	onpix	x-bar	y-bar	x2bar	y2bar	xybar	x2ybr	xy2br	x-ege	xegvy	y-ege	yegvx
3	11	4	8	2	1	13	5	4	12	10	7	0	8	3	6
4	10	5	7	3	5	10	9	4	7	4	8	3	7	6	11
5	9	6	5	3	10	3	4	6	12	4	10	3	8	7	10
4	6	6	6	6	6	6	9	5	3	6	4	8	7	8	9
4	6	6	4	4	10	6	7	5	6	7	4	8	5	2	5
4	5	4	8	2	1	14	5	4	12	10	6	0	8	2	6
4	6	4	4	3	7	6	9	7	7	7	7	2	8	8	9
4	9	5	7	4	7	7	3	12	9	6	8	0	8	8	8
8	12	8	6	4	10	6	3	9	9	2	5	4	6	4	10
4	10	5	8	4	6	10	6	5	9	7	3	2	10	4	7
6	7	8	5	5	7	8	2	7	10	5	9	4	7	3	7
7	9	8	4	3	5	9	3	4	13	9	9	5	8	0	8
7	12	6	6	3	8	9	7	5	14	4	4	4	10	4	7
5	10	7	8	4	7	7	3	11	11	6	8	1	8	7	8
3	8	3	6	3	7	7	12	1	6	6	8	5	8	0	8
4	9	6	6	7	9	6	4	4	6	7	7	7	9	8	6
2	8	3	6	2	2	11	4	5	11	10	8	0	8	2	7
2	4	4	3	2	6	7	2	7	11	7	9	2	8	4	8
6	10	9	7	7	8	7	5	6	9	5	6	3	7	7	10
3	4	4	6	3	6	9	9	8	7	5	7	2	8	9	10

No lettr attribute, use it to predict it is outlier or not!

# Attributes Description

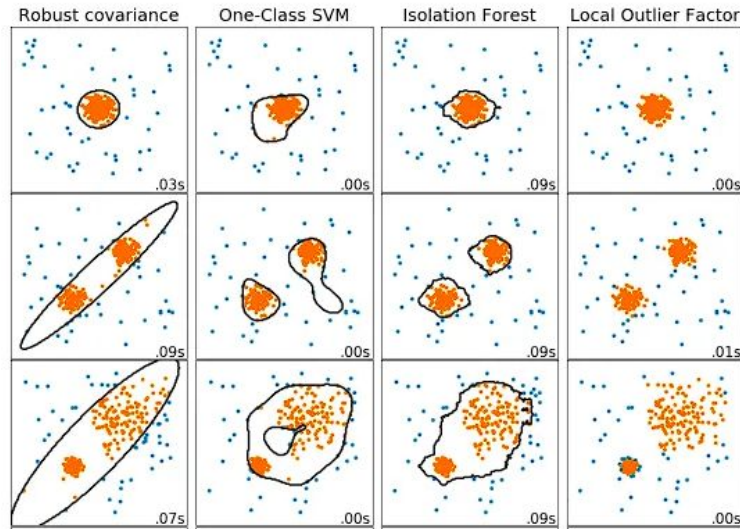
- lettr capital letter (26 values from A to Z)
- x-box horizontal position of box (integer)
- y-box vertical position of box (integer)
- width width of box (integer)
- high height of box (integer)
- onpix total # on pixels (integer)
- x-bar mean x of on pixels in box (integer)
- y-bar mean y of on pixels in box (integer)

# Attributes Description

- $x2bar$  mean x variance (integer)
- $y2bar$  mean y variance (integer)
- $xybar$  mean x y correlation (integer)
- $x2ybr$  mean of  $x * x * y$  (integer)
- $xy2br$  mean of  $x * y * y$  (integer)
- $x\text{-ege}$  mean edge count left to right (integer)
- $xegvy$  correlation of x-ege with y (integer)
- $y\text{-ege}$  mean edge count bottom to top (integer)
- $yegvx$  correlation of y-ege with x (integer)

# Method 1 - SVM

- Use OneClass SVM to learn a decision boundary.
- Find the suitable kernel space and parameters to fit the data.
- Convert the result of classification to the self-defined value.

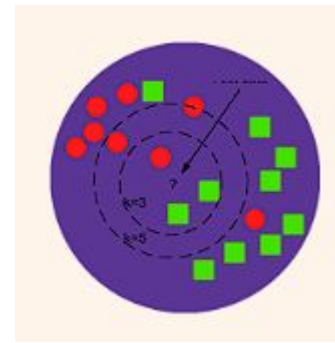


Overview of outlier detection methods



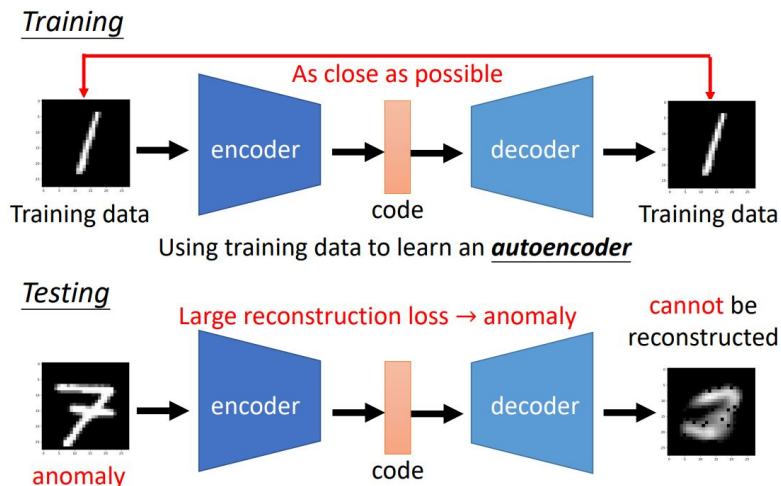
## Method 2 - KNN

- Assume that there are  $n$  clusters in training data.
- Assume that  $n$  is a small value
- Using K-means to calculate the  $n$  centroids of training data. Then use these  $n$  centroids to cluster the testing data.
- In the same cluster, the distance between inliers to centroid must be smaller than the distance between outliers to centroid.
- We can take the distance to centroids as the weight value for prediction.



## Method 3 - Autoencoder

- Using training data to train a AE or VAE
- Because the outliers cannot be reconstructed well, the MSE of outliers must be greater than inliers.
- We can take the reconstruction loss as the weight value for prediction.



## Methed 4 - Any reasonable way you can think

The key point is to make objects within the same group as similar as possible, and keeping those in different groups to be as dissimilar as possible.

HW3? No way!



No restrictions?



Let me think...



Ah, ha!



Compactness & Separation



# Kaggle Submission

- [Kaggle link](#)
- Display team name : <student ID>
  - team name error : -5%
- Submission format
  - A 1001\*2 .csv file, index start from 0. Outliers are any weight values that you define.(MSE, F1Loss, distance etc.)
  - Column name must be **id** and **outliers**.
  - [sample submission](#)
- There are one simple baseline and one strong baseline, beat them to get the higher score.

id	outliers
0	3.865861106
1	1.564198455
2	1.427000115
3	1.1940908
4	2.475497267
5	-0.0849644

# Kaggle Submission

- You can submit at most 5 times each day.
- The scoring metric is **auc score**.
- You can choose 2 of the submissions to be considered for the private leaderboard, or will otherwise default to the best public scoring submissions.

# Report Submission

Answer the following 3 questions:

1. Explain your implementation which get the best performance in detail.
2. Explain the rationale for using auc score instead of F1 score for binary classification in this homework.
3. Discuss the difference between semi-supervised learning and unsupervised learning.

Please answer the question in detail !

# Grading policy

- Kaggle (70%)
  - Basic score:
    - Over strong baseline : 55
    - Over simple bassline : 40
    - Under simple baseline : 25
  - Ranking score:
    - $15 - (15/N) * (\text{ranking} - 1)$ , N=numbers of people
  - Final Kaggle score depend on 30% public leaderboard 70% private leaderboard
- Source code and report (30%)
  - 10% for each quesiton

# E3 Submission

Submit your source code and report to E3 before 6/21 23:59.

No late submission !

- Format
  - source code : HW3\_<student ID>.py or HW3\_<student ID>.ipynb
  - report : HW3\_<student ID>.pdf

If you have any question about HW3, please feel free to contact with TA: GUAN-YI HO

through email [joy861106.cs11@nycu.edu.tw](mailto:joy861106.cs11@nycu.edu.tw)



Take Easy

