
# COMP9321:
# Data services engineering

## Week 9: Hierarchal Clustering and ML Model Evaluation

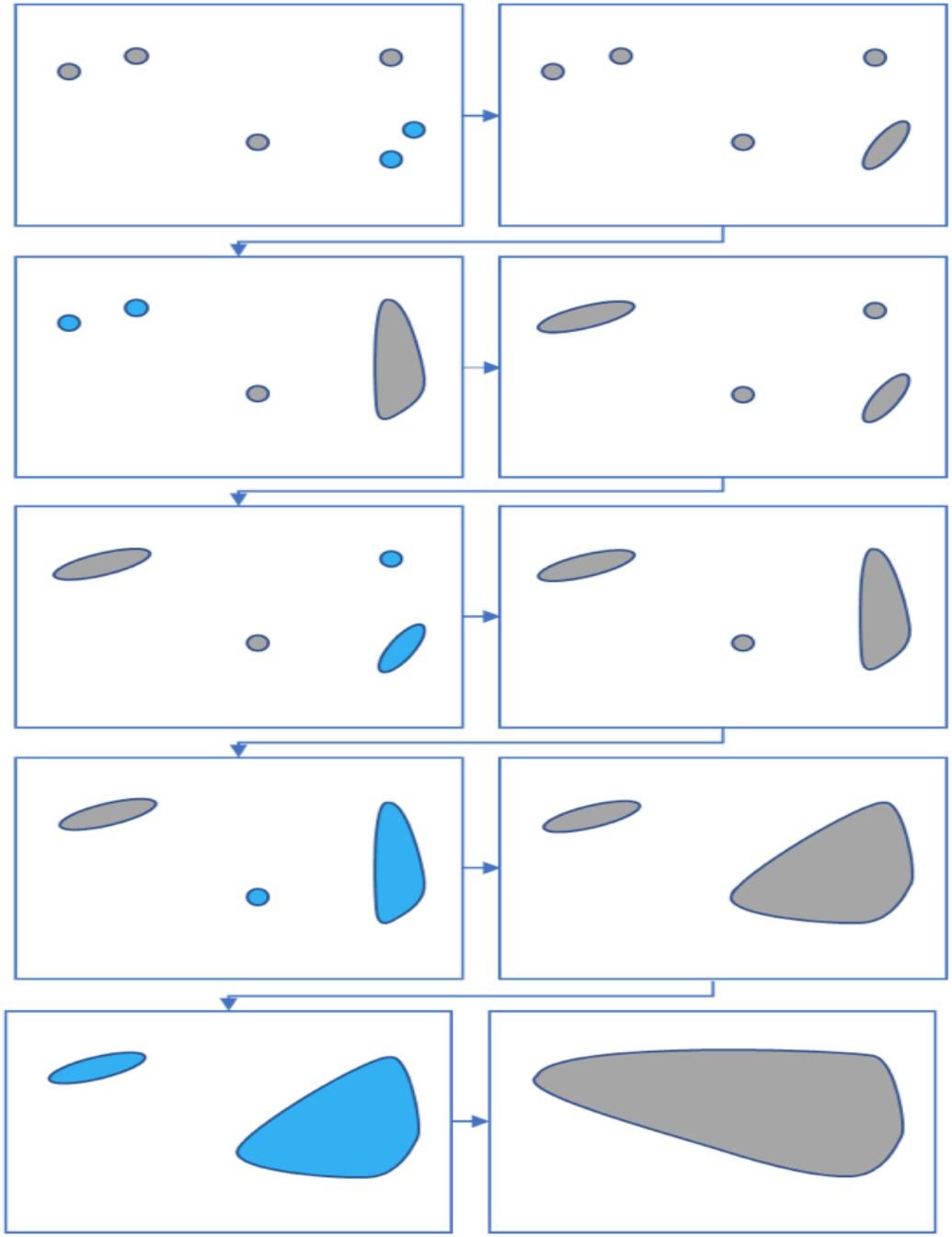
**Term 1, 2020**

**By Mortada Al-Banna, CSE UNSW**

层级划分

# Hierarchal Clustering

- What is it?
  - Unsupervised machine learning.
  - It is essentially building a hierarchy of clusters
- Types of Hierarchal Clustering
  - Agglomerative hierarchical clustering
  - Divisive Hierarchical clustering



# Linkage Criteria

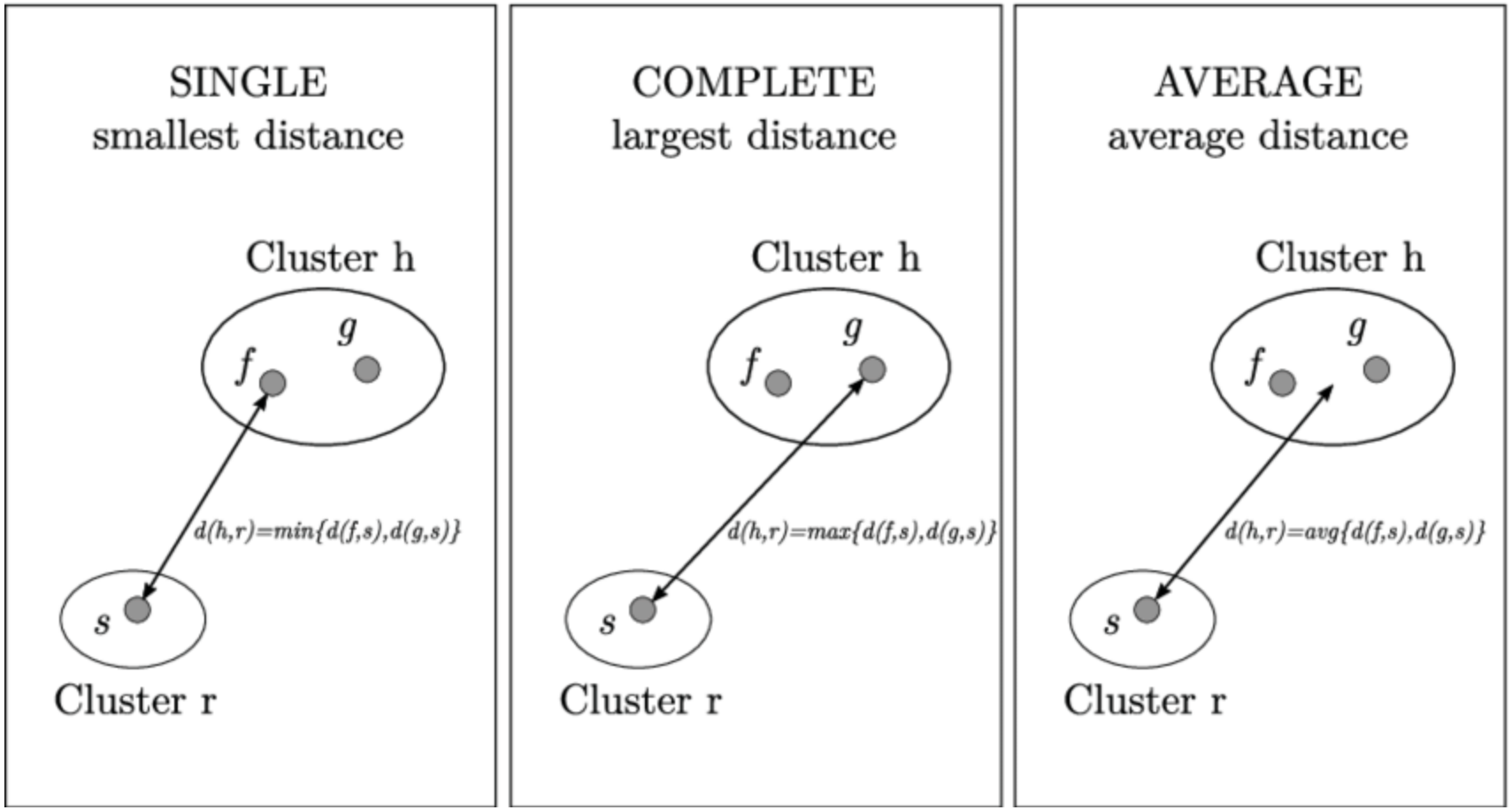
- It is necessary to determine from where distance is computed in cluster.
- Your options
  - ➢ It can be computed between the two most similar parts of a cluster (*single-linkage*)
  - ➢ the two least similar bits of a cluster (*complete-linkage*)
  - ➢ the center of the clusters (*mean* or *average-linkage*)
  - ➢ or some other criterion

从两个cluster里取距离最短的两个点当作cluster的距离

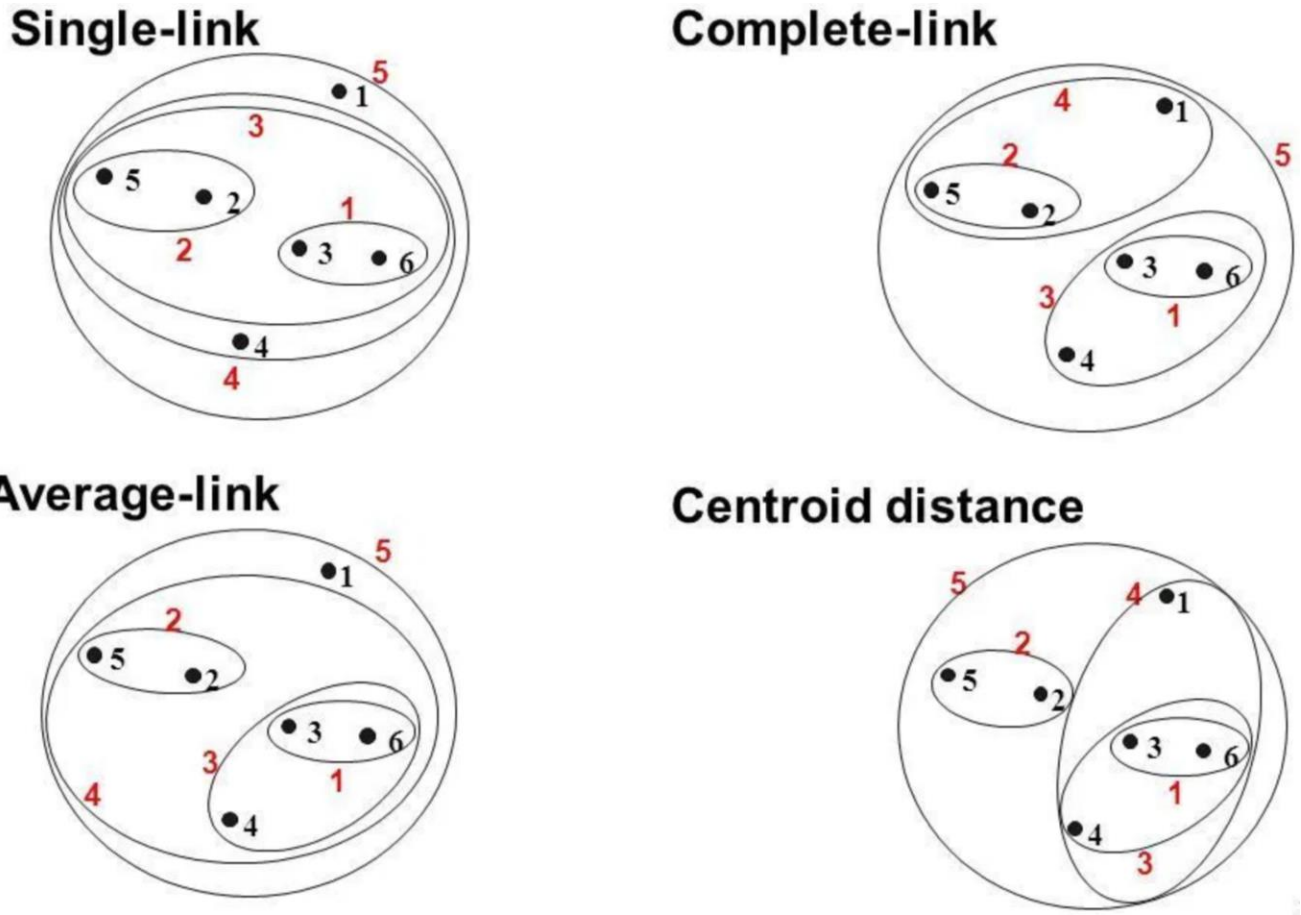
从两个cluster里取距离最长的两个点当作cluster的距离

从两个cluster里的两个中点当作cluster的距离

# Linkage Criteria



# Linkage Criteria Comparison

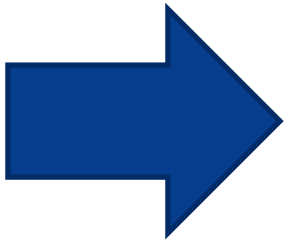


# Agglomerative Clustering Algorithm

1. Compute the proximity matrix
2. Let each data point be a cluster
3. Repeat: Merge the two closest clusters and update the proximity matrix
4. Until only a single cluster remains

# Agglomerative Clustering Example

| Student_ID | Marks |
|---|---|
| 1 | 10 |
| 2 | 7 |
| 3 | 28 |
| 4 | 20 |
| 5 | 35 |

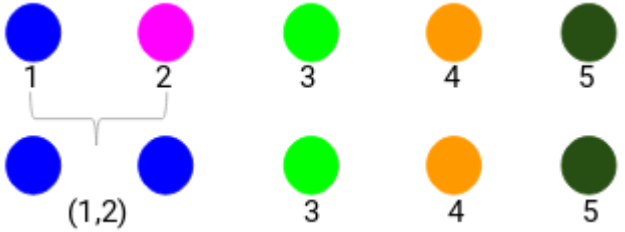


| ID | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| **1** | 0 | 3 | 18 | 10 | 25 |
| **2** | 3 | 0 | 21 | 13 | 28 |
| **3** | 18 | 21 | 0 | 8 | 7 |
| **4** | 10 | 13 | 8 | 0 | 15 |
| **5** | 25 | 28 | 7 | 15 | 0 |

**Proximity Matrix**

# Agglomerative Clustering Example

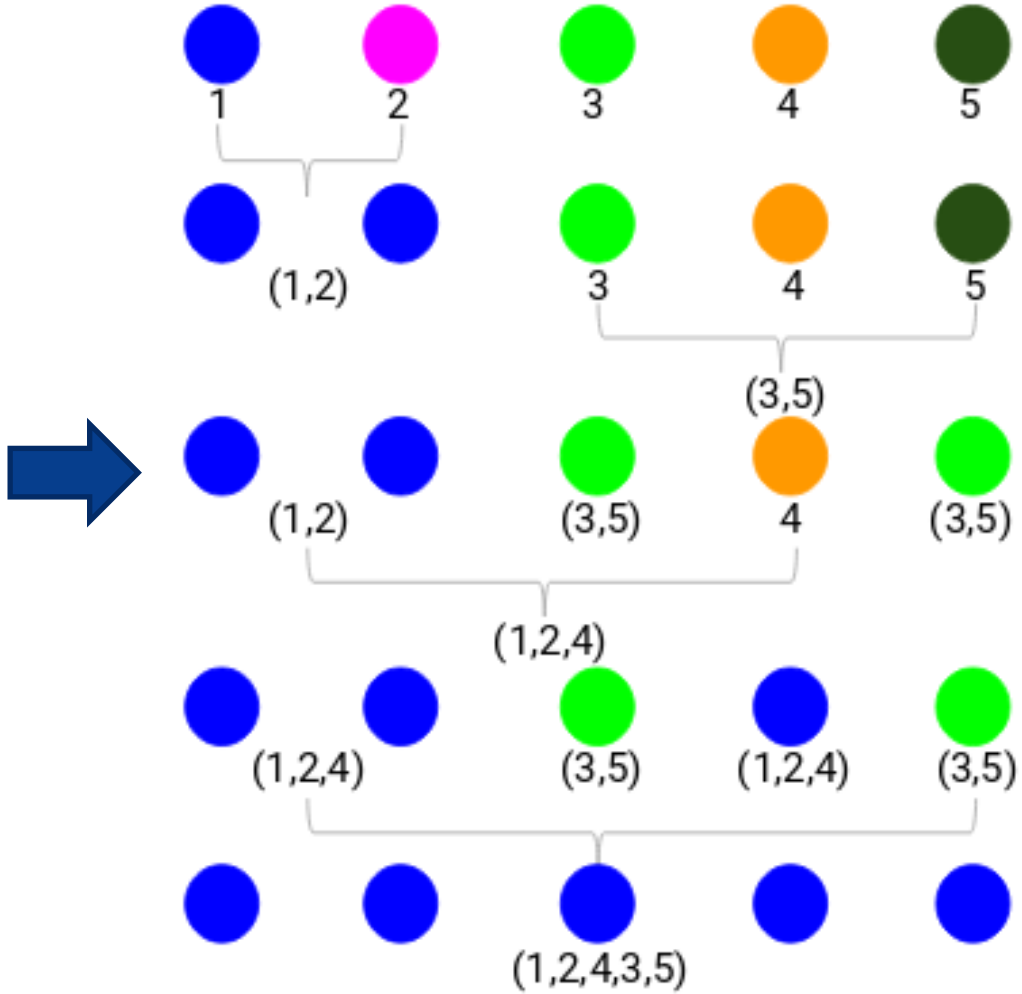

| ID | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | 0 | 3 | 18 | 10 | 25 |
| 2 | 3 | 0 | 21 | 13 | 28 |
| 3 | 18 | 21 | 0 | 8 | 7 |
| 4 | 10 | 13 | 8 | 0 | 15 |
| 5 | 25 | 28 | 7 | 15 | 0 |




| Student_ID | Marks |
|---|---|
| (1,2) | 10 |
| 3 | 28 |
| 4 | 20 |
| 5 | 35 |

# Agglomerative Clustering Example

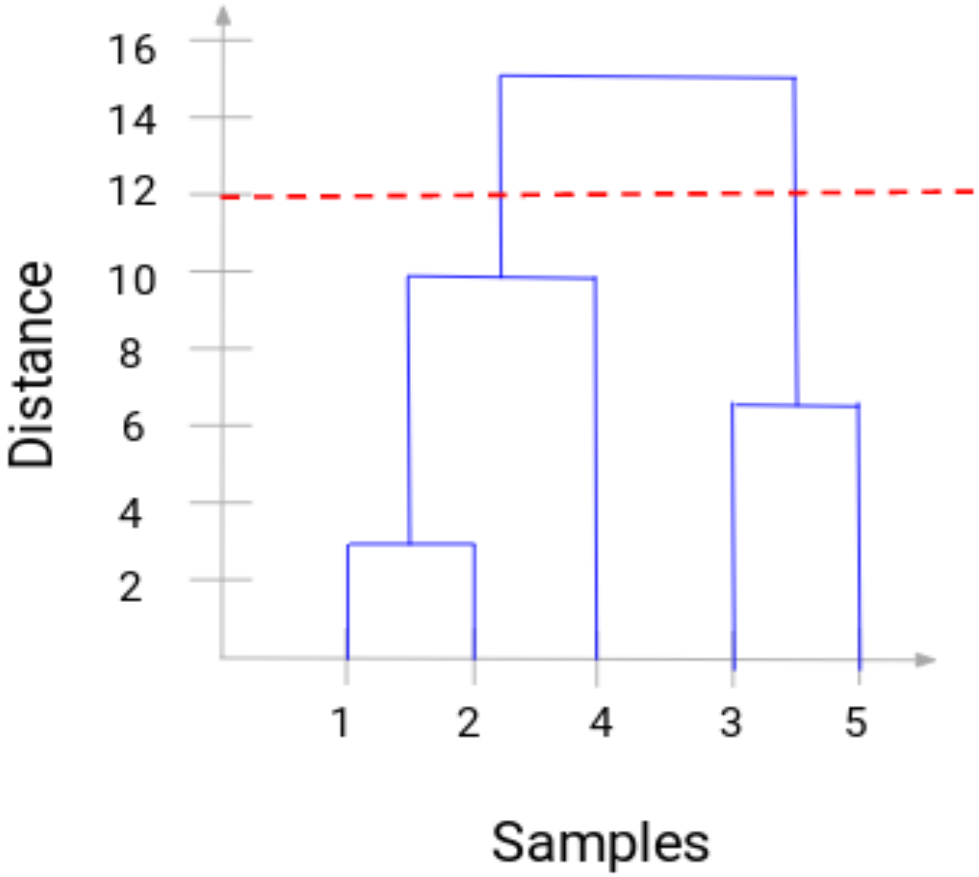
| ID | (1,2) | 3 | 4 | 5 |
|---|---|---|---|---|
| (1,2) | 0 | 18 | 10 | 25 |
| 3 | 18 | 0 | 8 | 7 |
| 4 | 10 | 8 | 0 | 15 |
| 5 | 25 | 7 | 15 | 0 |



# How Can we Choose the Number of Clusters?

- Using a Dendrogram
- A dendrogram is a tree-like diagram that records the sequences of merges or splits
- Whenever two clusters are merged, we will join them in this dendrogram and the height of the join will be the distance between these points
- We set a threshold distance and draw a horizontal line (try to set the threshold in such a way that it cuts the tallest vertical line)
- The number of clusters will be the number of vertical lines which are being intersected by the line drawn using the threshold

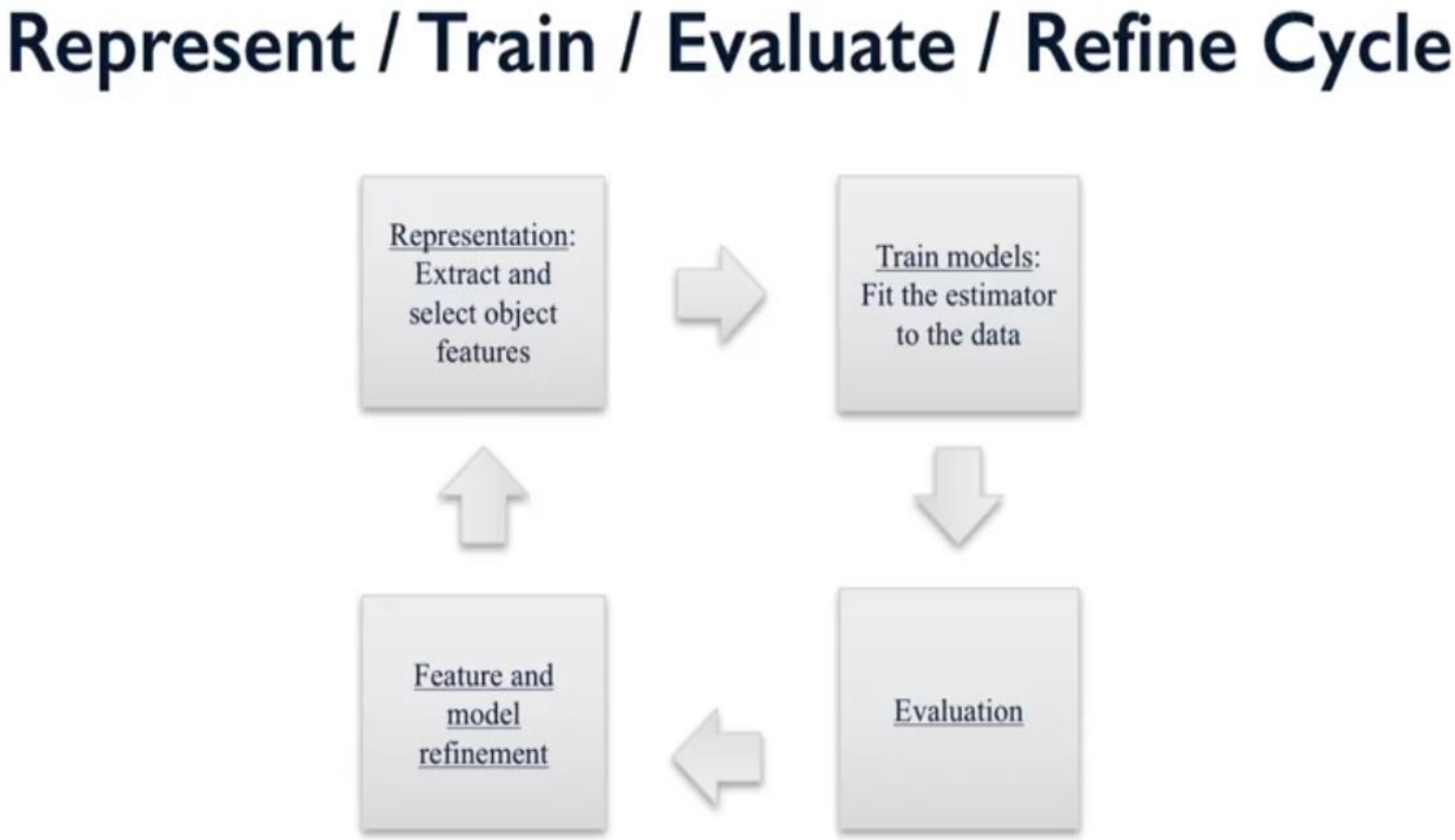
# How Can we Choose the Number of Clusters?



# Advantages and Disadvantages of Hierarchal Clustering

- Advantages
  - Easy to Implement
  - No Need to decide the number of clusters beforehand.
- Disadvantages
  - Not suitable for large datasets
  - Sensitive to Outliers
  - Initial Seeds have strong impact of final results
  - Linkage criteria and Distance measure are selected most of the time arbitrary.

# Refresher


Represent / Train / Evaluate / Refine Cycle
Representation: Extract and select object features
Train models: Fit the estimator to the data
Evaluation
Feature and model refinement

# Machine Learning Evaluation

- There are various metrics and methods to evaluate machine learning algorithms
- They differ according to the algorithm being supervised or unsupervised and they differ according to the task
- Let’s look at some of the metrics and concepts regarding evaluation

# Accuracy

- This is the simplest metric
- Number of correct predictions divided by the total number of predictions, multiplied by 100.

$$\text{Accuracy} = \frac{\#\text{correct predictions}}{\#\text{total instances}}$$

# Accuracy with Imbalanced Classes

- Suppose you have two classes:
  - The positive class
  - The negative class
- Out of 1000 randomly selected items, on average:
- One item belong to the positive class
- The rest of items (999 of them) belong to the negative class
- The Accuracy will be

$$\text{Accuracy} = \frac{\#\text{correct predictions}}{\#\text{total instances}}$$

# Accuracy with Imbalanced Classes

- When you build a classifier to predict the items (positive or negative), you may find out that the accuracy on the test set is 99.9%.
- Be aware that this is not an actually presentation of how good your classifier is.
- For comparison, if we have a “dummy” classifier that do not consider the features at all but rather blindly predict according  to the most frequent class

# Accuracy with Imbalanced Classes

- If we use the same dataset mentioned in the previous slide (the 1000 data instance with 999 negative and 1 positive). What do you think the accuracy of the dummy classifier would be?

**Answer**:

$$\text{Accuracy}_{\text{Dummy}} = 999/1000 = 99.9\%$$

- Hence the accuracy alone sometime not a good metric to measure how good the model is

# Dealing with Imbalanced Classes

- Data pre-processing
  - Random Under Sampling
  - Random Over Sampling
  - Cluster-Based Over Sampling
  - Synthetic Minority Over-sampling
- Select More suitable Metrics to Evaluate Imbalanced Classes
  - Precession and Recall
  - F1-Score
  - Log-Loss

# Precision and Recall

**FP: the classifier predict it positive, but it is actually negative!**

**FP: the classifier predict it negative, but it is actually positive!**

## Precision

**Precision** attempts to answer the following question:

> What proportion of positive identifications was actually correct?

Precision is defined as follows:

$$\text{Precision} = \frac{TP}{TP + FP}$$

**TP: True Positive**

**FP: False Positive**

**FN: False Negative**

## Recall

**Recall** attempts to answer the following question:

> What proportion of actual positives was identified correctly?

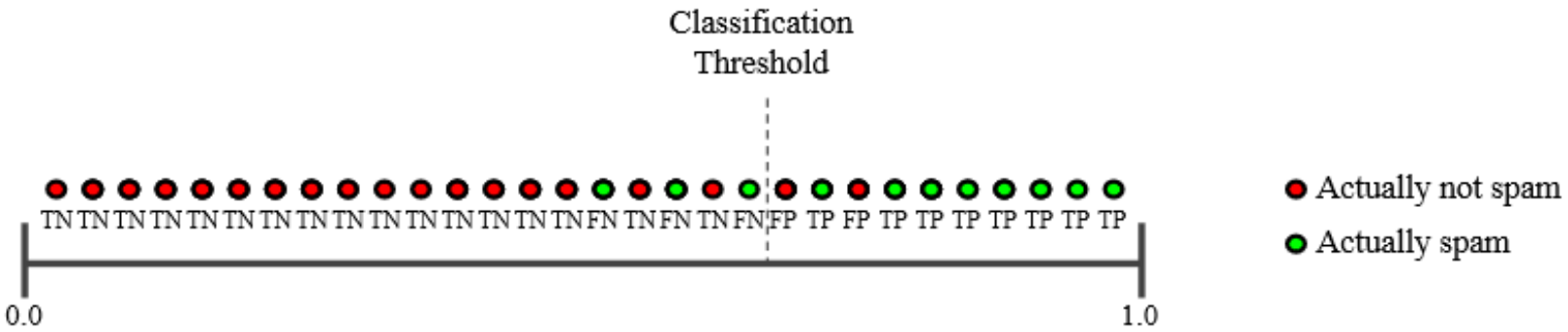
Mathematically, recall is defined as follows:

$$\text{Recall} = \frac{TP}{TP + FN}$$



https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall

# Precision and Recall




| | |
|---|---|
| True Positives (TP): 8 | False Positives (FP): 2 |
| False Negatives (FN): 3 | True Negatives (TN): 17 |

Precision measures the percentage of **emails flagged as spam** that were correctly classified—that is, the percentage of dots to the right of the threshold line that are green

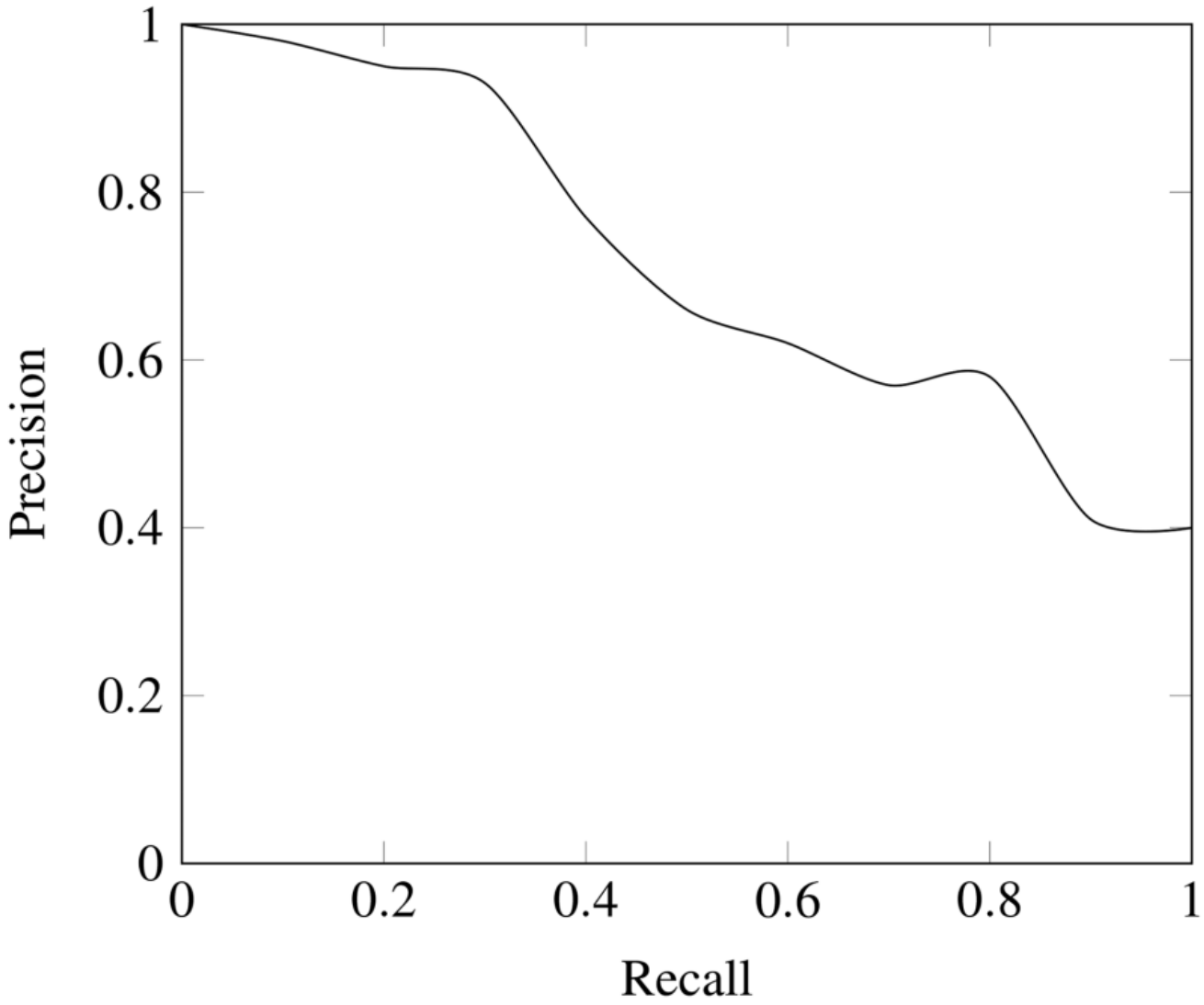
$$\text{Precision} = \frac{TP}{TP + FP} = \frac{8}{8 + 2} = 0.8$$

Recall measures the percentage of **actual spam emails** that were correctly classified—that is, the percentage of green dots that are to the right of the threshold line

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{8}{8 + 3} = 0.73$$


https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall

# Precision and Recall






https://towardsdatascience.com/beyond-accuracy-precision-and-recall-3da06bea9f6c

# F1 Score

- A metric which combines precision and recall
- Harmonic mean of precision and recall

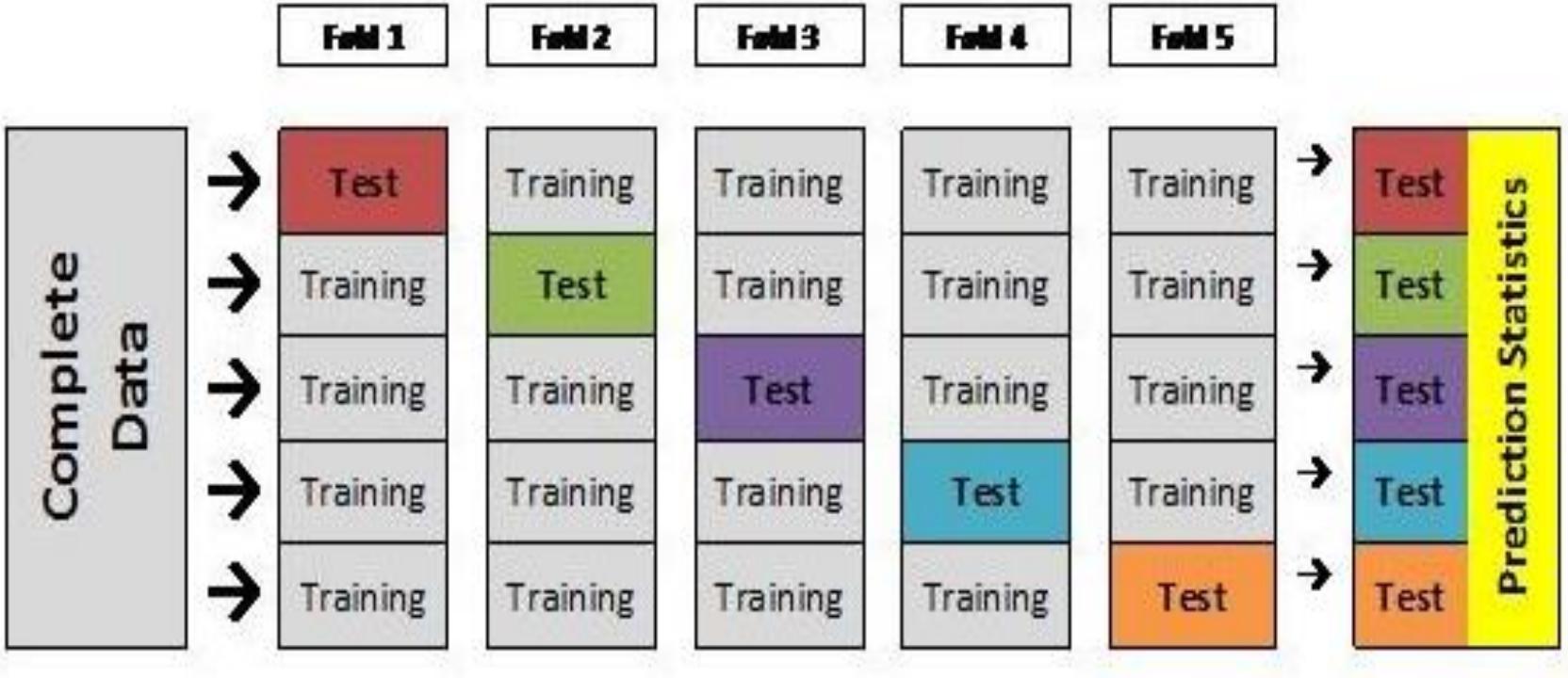
F1-score= 2*Precision*Recall/(Precision+Recall)

# Cross-validation

- Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample.
- The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into. As such, the procedure is often called k-fold cross-validation.
- When a specific value for k is chosen, it may be used in place of k in the reference to the model, such as k=5 becoming 5-fold cross-validation.


https://machinelearningmastery.com/k-fold-cross-validation/

# Cross Validation Examples (5-fold)






https://www.datasciencecentral.com/profiles/blogs/14-great-articles-about-cross-validation-model-fitting-and-select

# Stratified Cross-validation

| fruit_label | fruit_name |
|---|---|
| 1 | Apple |
| 1 | Apple |
| 1 | Apple |
| 1 | Apple |
| 1 | Apple |
| 2 | Mandarin |
| ... | ... |
| 3 | Orange |
| ... | ... |
| 4 | Lemon |
| 4 | Lemon |
| 4 | Lemon |
| 4 | Lemon |
| 4 | Lemon |

(Folds and dataset shortened for illustration purposes.)

Example has 20 data samples
= 4 classes with 5 samples each.

5-fold CV: 5 folds of 4 samples each.

Fold 1 uses the first 20% of the dataset as the test set, which only contains samples from class 1.

Classes 2, 3, 4 are missing entirely from test set and so will be missing from the evaluation.



https://www.coursera.org/learn/python-machine-learning

# Stratified Cross-validation

- Stratification is a technique where we rearrange the data in a way that each fold has a good representation of the whole dataset
- It forces each fold to have at least m instances of each class. T
- his approach ensures that one class of data is not overrepresented especially when the target variable is unbalanced.



https://medium.com/datadriveninvestor/k-fold-and-other-cross-validation-techniques-6c03a2563f1e

# Useful Resources

https://towardsdatascience.com/understanding-the-concept-of-hierarchical-clustering-technique-c6e8243758ec

https://heartbeat.fritz.ai/introduction-to-machine-learning-model-evaluation-fa859e1b2d7f

https://machinelearningmastery.com/tour-of-evaluation-metrics-for-imbalanced-classification/

https://www.analyticsvidhya.com/blog/2017/03/imbalanced-data-classification/

https://medium.com/james-blogs/handling-imbalanced-data-in-classification-problems-7de598c1059f



https://medium.com/datadriveninvestor/k-fold-and-other-cross-validation-techniques-6c03a2563f1e

# Q&A