Lab 7
Clustering

Oct 26, 2023

Department of Computing

The Hong Kong Polytechnic University

Today's Arrangement

- Assignment 3 Check
- Clustering basics
- Exercises
- Assignment 4

Assignment 3 Check

Assignment 3

As an expert in artificial intelligence, banks are looking to your expertise to predict credit card approvals. They provide you with a credit card database and your task is to determine whether the credit card application should be approved based on the variables within it.

To maintain confidentiality, all variable names and values have been anonymized and replaced with symbols.

Variable names	Role	Type	Value
Feature 1	Feature	Continuous	-
Feature 2	Feature	Continuous	-
Feature 3	Feature	Continuous	-
Feature 4	Feature	Continuous	-
Feature 5	Feature	Continuous	-
Feature 6	Feature	Continuous	-
Feature 7	Feature	Categorical	0,1
Feature 8	Feature	Categorical	0,1
Feature 9	Feature	Categorical	0,1
Feature 10	Feature	Categorical	0,1
Feature 11	Feature	Categorical	1,2,3
Feature 12	Feature	Categorical	1,2,3
Feature 13	Feature	Categorical	1,2,3,4,5,6,7,8,9
Feature 14	Feature	Categorical	1,2,3,4,5,6,7,8,9,10,11,12,13,14
Label	Label	Categorical	0,1 (0 for non-approval and 1 for approval)

Assignment 3

The bank required the use of decision tree-based models, emphasizing the importance of interpretability in the decision-making process. Try using a decision tree-based model for the best results!

You will train and validate a decision tree-based model using 590 samples in the public dataset. Your results will then be tested on 100 samples of the private dataset.

Hint 1: Cross-validation is important.
Hint 2: Consider preprocessing and feature engineering if it benefits your model.
Hint 3: Optimize hyperparameters for improved performance.
Hint 4: Other techniques like pre-pruning and post-pruning can be applied.

Assignment 3

- Please download the public dataset from https://drive.google.com/file/d/1gQ_hA5DLMYQHcqGmkSaq2w-Sm-OLAKfZ/view?usp=sharing
- Please download the private dataset from https://drive.google.com/file/d/1s1xhpfWKWeACf3SiPAmEhCs8dfwmSVEP/view?usp=sharing
- The above datasets are also available from Blackboard.
- Please use the following Python template for submission. (You can copy the code below from lab6-Exercise.ipynb)
- Your results will be evaluated on 100 samples in the private dataset (The labels will released in Lab 7).

Assignment 3

```
def read_data_from_csv(path):
    """Load datasets from CSV files.
    Args:
        path (str): Path to the CSV file.
    Returns:
        X (np.ndarray): Features of samples.
        y (np.ndarray): Labels of samples, only provided in the public datasets.
    """
    assert os.path.exists(path), f'File not found: {path}!'
    assert os.path.splitext(path)[
        -1] == '.csv', f'Unsupported file type {os.path.splitext(path)[-1]}!'

    data = pd.read_csv(path)
    column_list = data.columns.values.tolist()

    if 'Label' in column_list:
        # for the public dataset, label column is provided.
        column_list.remove('Label')
        X = data[column_list].values
        y = data['Label'].astype('int').values
        return X, y
    else:
        # for the private dataset, label column is not provided.
        X = data[column_list].values
        return X
```

```
X_public, y_public = read_data_from_csv('assignment_3_public.csv')
print('Shape of X_public:', X_public.shape)   # n_sample, m_feature
(590, 14)
print('Shape of y_public:', y_public.shape)  # n_sample (590,)

'''
CODE HERE!
'''

X_private = read_data_from_csv('assignment_3_private.csv')
print('Shape of X_private:', X_private.shape)   # k_sample, m_feature
(100, 14)

import numpy as np

# remove and make your own predictions.
preds = np.full(len(X_private), -1,
                dtype=int)
'''
CODE HERE!
e.g.,
preds = np.full(len(X_private), -1, dtype=int)
'''

submission = pd.DataFrame({'Label': preds})
submission.to_csv('assignment_3.csv', index=True, index_label='Id')
```

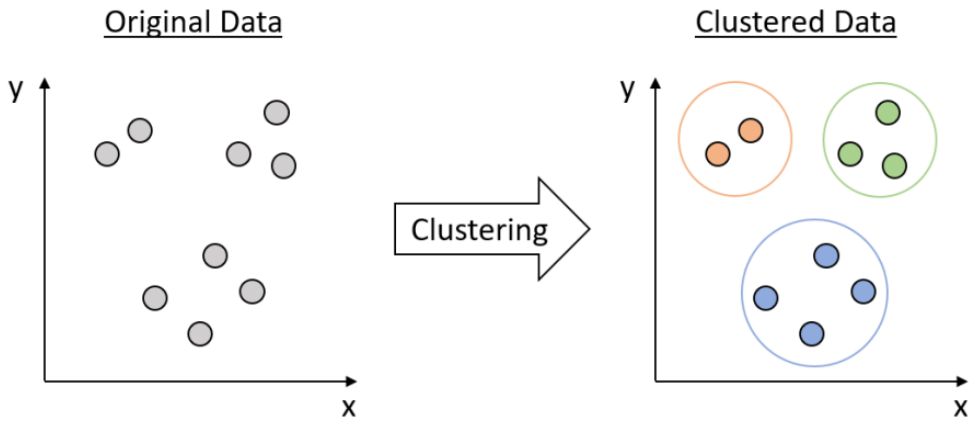
Assignment 3 Check

- Check your answer:
  - **Assignment3_Answer.ipynb**: available on Blackboard or https://colab.research.google.com/drive/1fC9msmE-3QElMIuEfPLE9z2F0ZpC9Qnb
  - **Private dataset**: available on Blackboard
- Some suggestions/comments:
  - You may need to revise the generated code (by hand or through multiple interactions with ChatGPT).
  - Utilize cross-validation to reduce the variance of the performance estimate and use more data for training.
  - Grid search can be used to find the optimal combination of hyper-parameters.
  - Decision trees can naturally handle missing values (by making splits based on the available data) and outliers (since the outliers are segregated into separate nodes, their impact on the overall model is lowered).
  - Pre-pruning and post-pruning: https://www.kaggle.com/code/arunmohan003/pruning-decision-trees-tutorial

Clustering basics

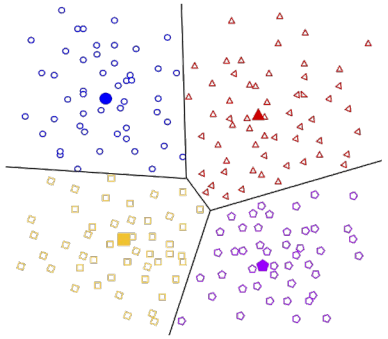
Clustering basics

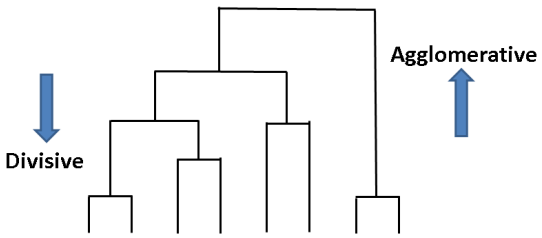
- Clustering
  - An unsupervised learning task, Requires data without labels
  - Partition a given dataset into groups based on specified features so that there is
    - High intra-cluster similarity
    - Low inter-cluster similarity



Clustering basics

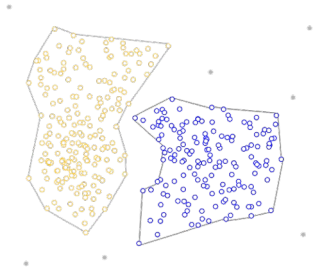
- Types of clustering
  - Centroid-based Clustering
    - Clusters are formed by the closeness of data points to the centroid of clusters.
    - K-Means, K-Medoids (using an actual point instead of the mean point as the center of a cluster), etc.
    - K-Means Online Demo: https://user.ceng.metu.edu.tr/~akifakkus/courses/ceng574/k-means/
  - Hierarchical Clustering
    - Build nested clusters by merging or splitting them recursively.
    - Two types: Divisive (Top down) and Agglomerative (Bottom up)



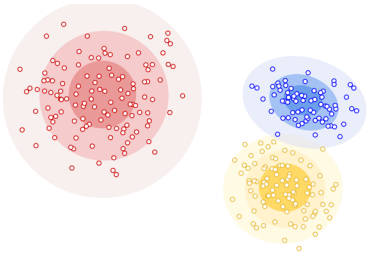


Clustering basics

- Types of clustering
  - Density-based Clustering
    - Identifying “dense” clusters of points, allowing it to learn clusters of arbitrary shape and identify outliers in the data
    - DBSCAN, etc.



  - Distribution-based Clustering
    - Assumes data is composed of distributions, such as Gaussian distributions.



  - ……

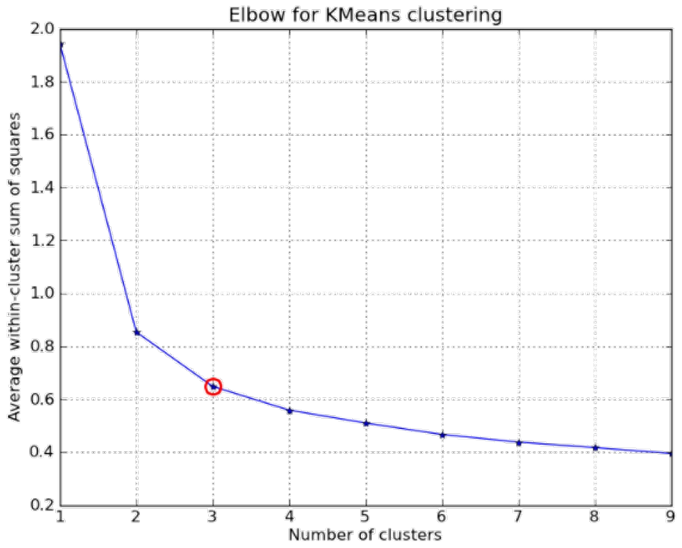
K-Means with scikit-learn

- sklearn.cluster.Kmeans()
  - Key Parameters:
    - n_clusters: The number of clusters to form (the number of centroids).
    - n_init: Number of times the k-means algorithm is run with different centroid seeds. (Run K-means with a large value of n_init, such as 20 or 50 to avoid getting stuck in an undesirable local optimum.)
    - max_iter: Maximum number of iterations of the k-means algorithm for a single run.

Reference: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

K-Means with scikit-learn

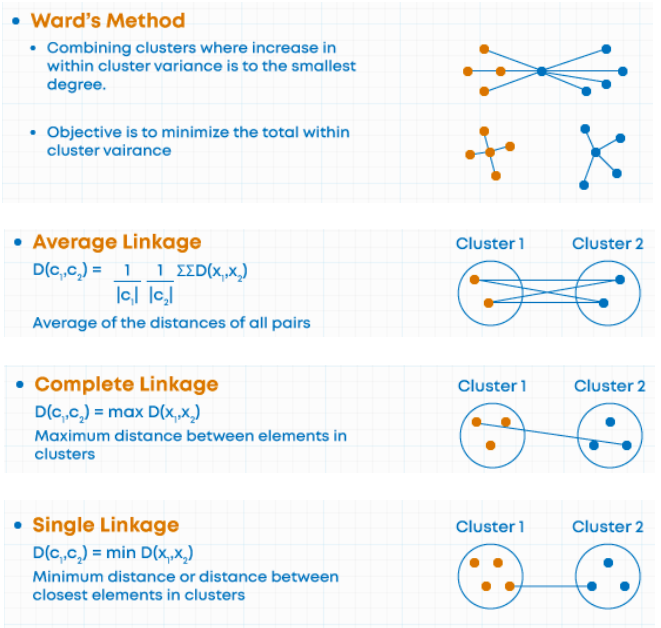
- Find the optimal number of clusters:
  - Elbow method:
    - As the number of clusters increases, the variation within each cluster decreases, but at some point (the elbow), the improvement becomes negligible.



  - Silhouette score: $$s(i) = \frac{b(i) - a(i)}{\max\left(a(i), b(i)\right)}$$
    - a: The mean distance between a sample and all other points in the same class.
    - b: The mean distance between a sample and all other points in the next nearest cluster.
    - A high value indicates that the object is well-matched to its own cluster and poorly matched to neighboring clusters.
    - Best: 1; Worst: -1

Hierarchical Clustering with scikit-learn

- sklearn.cluster.AgglomerativeClustering()
  - Recursively merges pair of clusters of sample data.
  - Key Parameters:
    - n_clusters: the number of clusters to find.
    - linkage: {“ward”, “complete”, “average”, “single”}
      The pair of clusters that minimize this criterion will be merged.
      - Ward Linkage: minimizes the variance of the clusters being merged.
      - Average Linkage: uses the average of the distances of each observation of the two sets.
      - Complete (Maximum) Linkage: uses the maximum distances between all observations of the two sets.
      - Single Linkage: uses the minimum of the distances between all observations of the two sets.



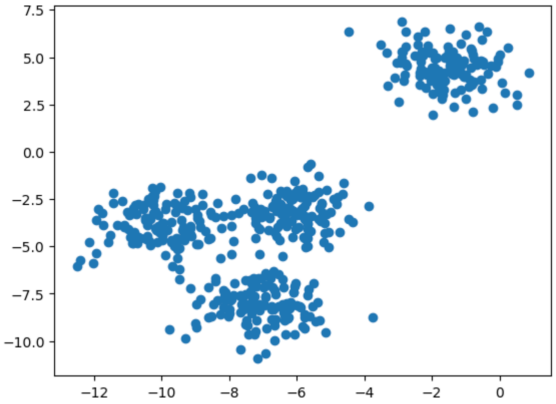
Reference: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html

Exercises

Exercise 1: K-Means

- Use K-Means to cluster the given dataset generated by the code.

```
from sklearn.datasets import make_blobs
X, y = make_blobs(
    n_samples=500,
    n_features=2,
    centers=4,
    cluster_std=1,
    center_box=(-10.0, 10.0),
    shuffle=True,
    random_state=1,
)
```



Requirements:

- Use different numbers of clusters.
- Visualize each clustering result.
- Choose the optimal number of clusters. (Elbow method or Silhouette score)

Exercise 1: K-Means

- Example prompt (Silhouette score):

Write a Python script to Use K-Means to cluster the given dataset generated by the code.

```
from sklearn.datasets import make_blobs
X, y = make_blobs(
   n_samples=500,
   n_features=2,
   centers=4,
   cluster_std=1,
   center_box=(-10.0, 10.0),
   shuffle=True,
   random_state=1,
)
```

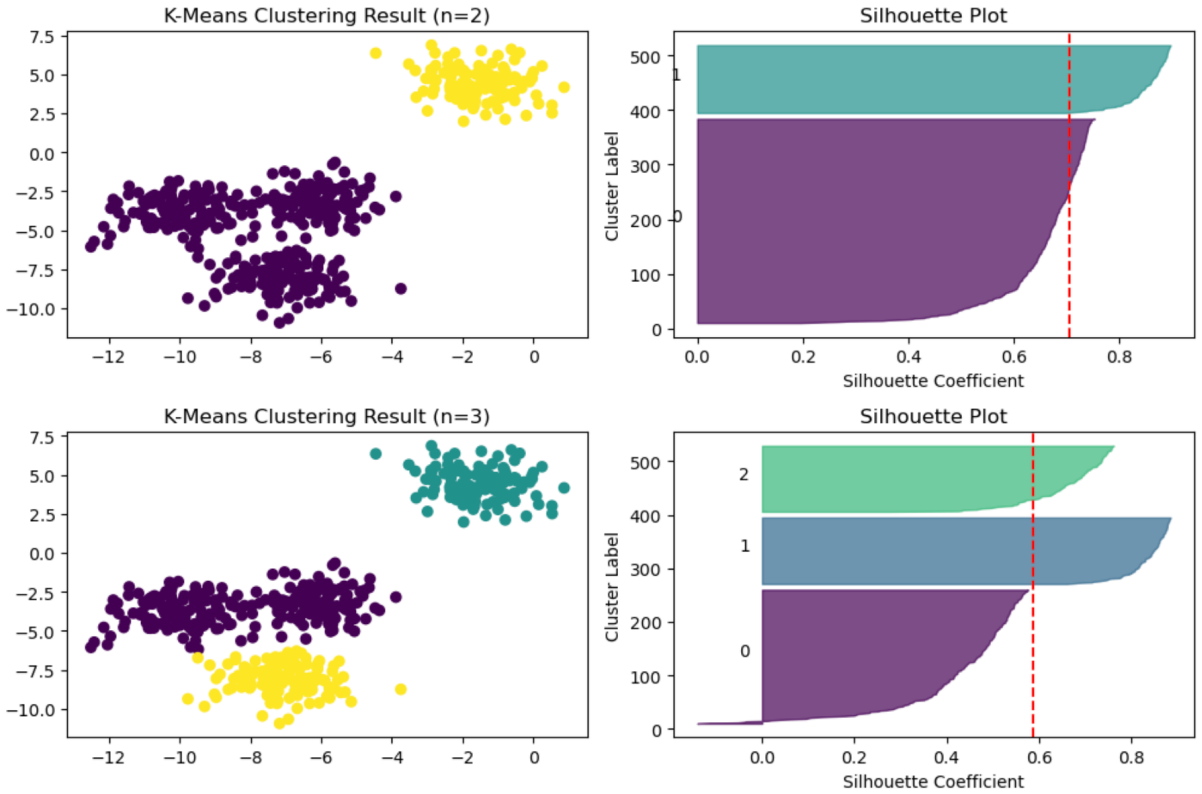
Requirements:

- Use different numbers of clusters.
- Visualize each clustering result.
- Draw a silhouette plot for each result.
- Choose the optimal number of clusters.

Exercise 1: K-Means

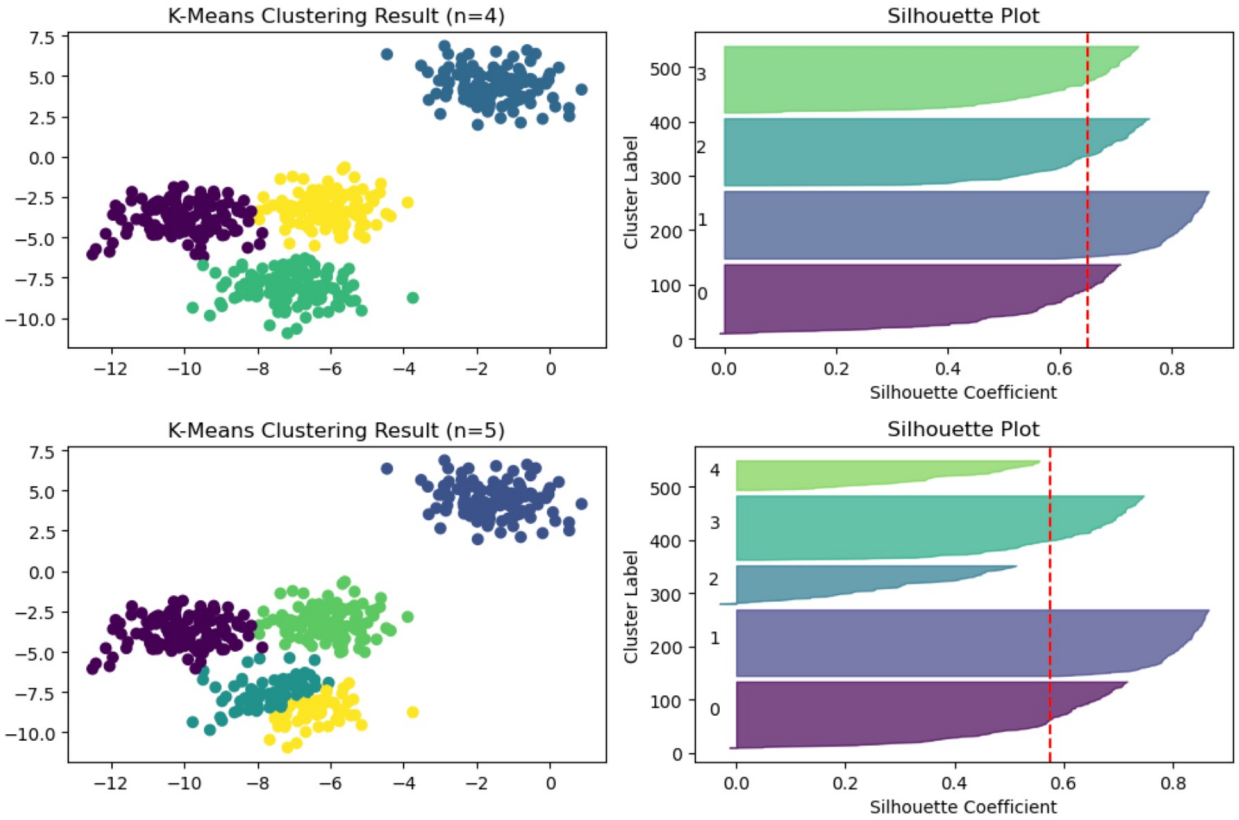
- Resulting clusters and silhouette plots

```
Optimal number of clusters: 2
```



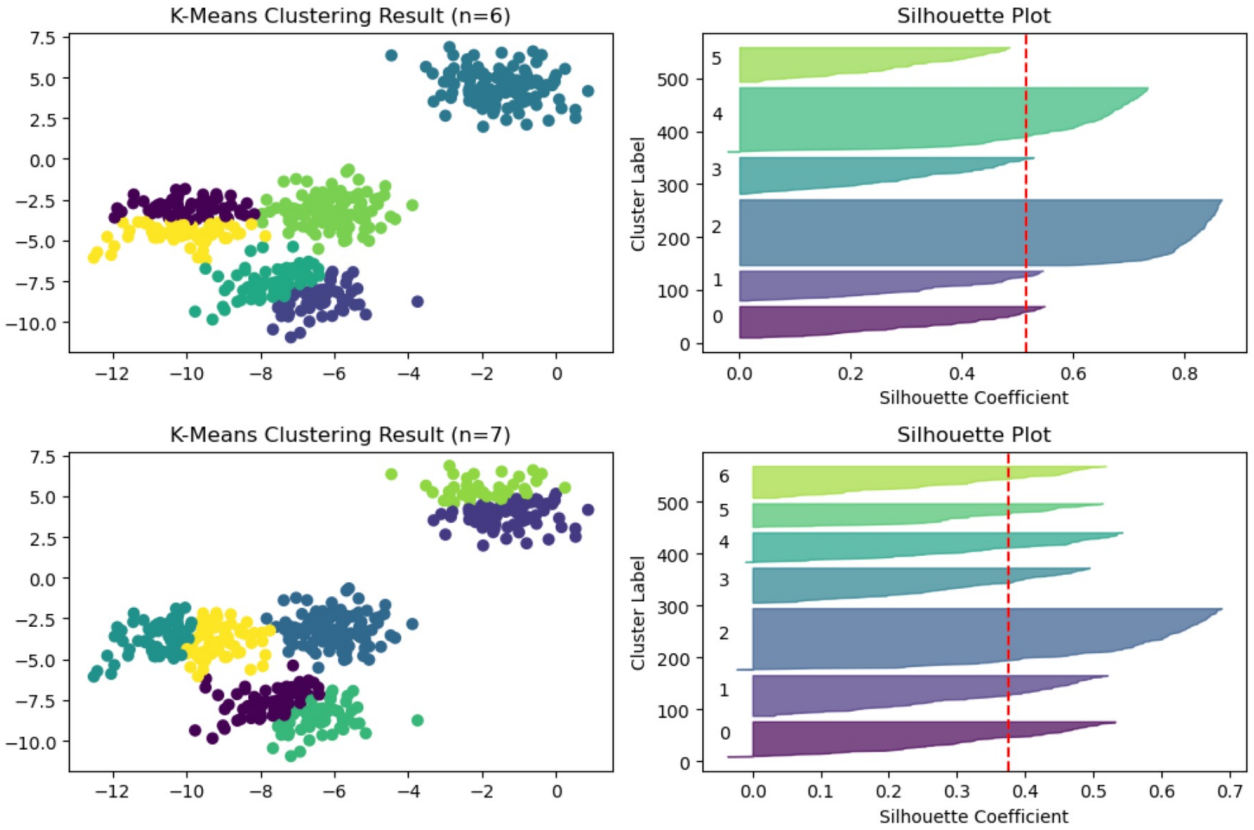
Exercise 1: K-Means

- Resulting clusters and silhouette plots



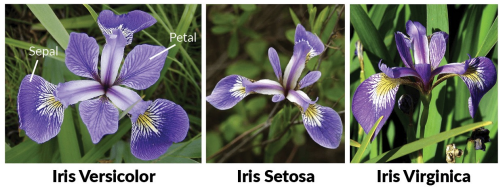
Exercise 1: K-Means

- Resulting clusters and silhouette plots



Exercise 2: Hierarchical Clustering

- Dataset: the iris dataset
  - 4 attributes and 150 samples. Each sample is labeled as one of the three types of Iris flowers.
- Tasks:
  1. Conduct hierarchical clustering on the Iris dataset. Ignore the labels and just use the attributes to cluster the dataset.
  2. Try different linkage criterions for the hierarchical clustering.
  3. Compare the results of different linkage criterions with the original labels to see which one performs better in this scenario.
  4. Visualize the resulting cluster hierarchies.



Iris Versicolor | Iris Setosa | Iris Virginica

Hint: Try to find out the answer to the below question first.
What is the measure of the similarity between two data clusterings?

Exercise 2: Hierarchical Clustering

- Example prompt

Write a Python script to Conduct hierarchical clustering on the Iris dataset. Follow the below steps:

1. Load the Iris dataset from sklearn

2. Ignore the labels and use the attributes to cluster the dataset into three clusters. Try different linkage criterions for the hierarchical clustering.

3. Use the **adjusted Rand score** to compare the clustering results with the original labels to see which linkage criterion performs better in this scenario.

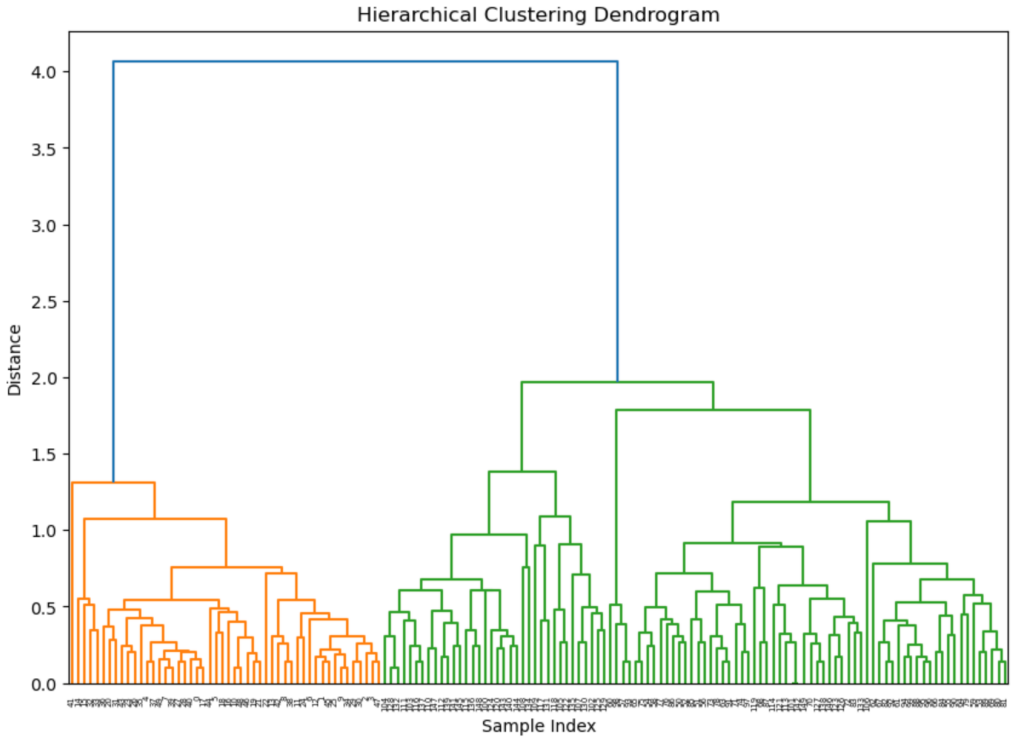
4. Visualize the resulting clustering hierarchy with the best linkage criterion.

Learn more about Rand index and Adjusted Rand index : https://en.wikipedia.org/wiki/Rand_index

Exercise 2: Hierarchical Clustering

- Example result

```
ARI score with ward linkage: 0.7311985567707746
ARI score with complete linkage: 0.6422512518362898
ARI score with average linkage: 0.7591987071071522
Best linkage criterion: average
```



Assignment 4

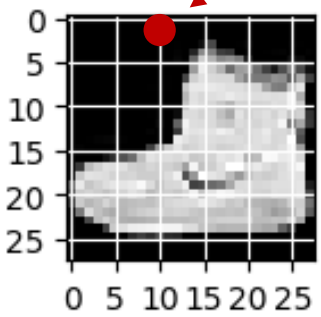
Assignment 4

As an artificial intelligence expert, a fashion company has invited you to help cluster the clothing images. You will be provided with a dataset with each sample corresponding to an image. Each image measures 28 pixels in height and 28 pixels in width, totaling 784 pixels.

These images are stored in a table, representing the pixels across 784 columns. Each column has a single pixel value, which is an integer ranging from 0 (black) to 255 (white).

To locate a pixel on the image, suppose we have the pixel at row j and column k of a 28×28 matrix; we can find its pixel value at the i-th column of the table, where $i = j\times28 + k$ where j and k are integers between 0 and 27.

$[j, k] = [1, 9]$



$i = 1 \times 28 + 9 = 37$

Pixel 37
0

Assignment 4

Your task is to use clustering algorithms to cluster these images. For example, you may find some images belonging to the cluster of pants, while others may be T-shirts, etc.

In the training stage, you will be provided with a public dataset comprising 3000 samples. The goal is to cluster these samples and assign the clustering results to each sample. In the test stage, your predictions will be evaluated using the adjusted rand score as the final performance. Please note that using any pre-trained models is not allowed, nor is the use of any form of external data permitted.

Hint: The number of clothing clusters is unknown. You need to determine the cluster number via analysis.
Hint: You can try various clustering algorithms to find the most suitable one.
Hint: You can use internal evaluation metrics, like the Davies-Bouldin Index, Dunn Index, and Silhouette Coefficient, to fine-tune hyperparameters.

Assignment 4

- Please download the public dataset from https://drive.google.com/file/d/1XJ0D1L4K-urd83nwCIB4BT75c2rSPTaj/view?usp=share_link or from Blackboard.
- Please use the following Python template for submission (You can copy the code below from lab7-Exercise.ipynb).

```
def read_data_from_csv(path):
    """Load datasets from CSV files.
    Args:
        path (str): Path to the CSV file.
    Returns:
        X (np.ndarray): Features of samples.
        y (np.ndarray): Labels of samples, only provided in the public datasets.
    """
    assert os.path.exists(path), f'File not found: {path}!'
    assert os.path.splitext(path)[
        -1] == '.csv', f'Unsupported file type {os.path.splitext(path)[-1]}!'

    data = pd.read_csv(path)
    column_list = data.columns.values.tolist()

    if 'Label' in column_list:
        # for the public dataset, label column is provided.
        column_list.remove('Label')
        X = data[column_list].values
        y = data['Label'].astype('int').values
        return X, y
    else:
        # for the private dataset, label column is not provided.
        X = data[column_list].values
        return X
```

```
X_public = read_data_from_csv('assignment_4_public.csv')
print('Shape of X_public:', X_public.shape)

# remove and make your own predictions.
preds = np.full(len(X_public), -1,
                dtype=int)

'''
CODE HERE!
'''

submission = pd.DataFrame({'Label': preds})
submission.to_csv('assignment_4.csv', index=True, index_label='Id')
```